Emilio Corchado
Marek Kurzyński
Michał Woźniak (Eds.)

LNAI 6679

Hybrid Artificial Intelligent Systems

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

2 Part II

Springer

Lecture Notes in Artificial Intelligence 6679

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 II

Springer

Volume Editors

Emilio Corchado
University of Salamanca
Faculty of Biology
BISITE Research Group
Salamanca, Spain
E-mail: escorchado@usal.es

Marek Kurzyński
Michał Woźniak
Wroclaw University of Technology
Faculty of Electronics
Department of Systems and Computer Networks
Wroclaw, Poland
E-mail: {marek.kurzynski,michal.wozniak}@pwr.wroc.pl

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21221-5 e-ISBN 978-3-642-21222-2
DOI 10.1007/978-3-642-21222-2
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

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 6$^{\text{th}}$ 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 Zmyślony, 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)
Michal Wozniak	Wroclaw University of Technology (Poland)

Program Committee

Emilio Corchado	University of Salamanca (Spain) (PC Co-chair)
Michał Woźniak	Wroclaw 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)
Estefania 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 Pontifícia de Salamanca (Spain)
John Beasley	Brunel University (UK)
Bruno Baruque	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	Wroclaw 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 Chyzhyk	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)
Guiomar 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	Wroclaw 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	Wroclaw 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 Braunschweig/Wolfenbuettel (Germany)
Andreas König	University of Kaiserslautern (Germany)
Mario Köppen	Kyushu Institute of Technology (Japan)
Rudolf Kruse	Otto-von-Guericke-Universität Magdeburg (Germany)
Bernadetta Kwintiana	Universität Stuttgart (Germany)
Dario Landa-Silva	University of Nottingham (UK)
Soo-Young Lee	Brain Science Research Center (Korea)
Lenka Lhotská	Czech Technical University in Prague (Czech Republic)
Hailin Liu	Guangdong University of Technology (China)
Otoniel López Granado	Universidad Autónoma de Madrid (Spain)
Karmele López de Ipiña	University of the Basque Country (Spain)
Oscar Luaces	Universidad de Oviedo (Spain)
Teresa Ludermir	Universidade Federal de Pernambuco (Brazil)
Julián Luengo	University of Granada (Spain)
Wenjian Luo	University of Science and Technology of China (China)
Núria Macià	Universitat Ramon Llull (Spain)
Kurosh Madani	University of Paris-Est Creteil (France)
Ana Maria Madureira	Instituto Politécnico do Porto (Portugal)
Roque Marin	University of Murcia (Spain)
Yannis Marinakis	Technical University of Crete (Greece)
Jose Luis Marroquin	CIMAT (Mexico)
José Fco. Martínez-Trinidad	INAOE (Mexico)
José Luis Martínez	University of Castilla La Mancha (Spain)
Jacinto Mata	University of Huelva (Spain)
Giancarlo Mauri	University of Milano-Bicocca (Italy)
Dawn Medlin	Appalachian State University (USA)
David Meehan	Dublin Institute of Technology (Ireland)
Gerardo M. Méndez	Instituto Tecnológico de Nuevo León (Mexico)
Abdel-Badeeh M. Salem	Ain Shams University (Egypt)
Masoud Mohammadian	University of Canberra (Australia)
José Manuel Molina	University Carlos III of Madrid (Spain)
Claudio Moraga	European Centre for Soft Computing (Spain)
Marco Mora	Universidad Católica del Maule (Chile)
Eduardo Morales	INAOE (Mexico)
Ramón Moreno	Universidad del País Vasco (Spain)
Juan Álvaro Muñoz Naranjo	Universidad de Almería (Spain)
Dimitris Mourtzis	University of Patras (Greece)
Susana Nascimento	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)
Michal 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 Zopounidis	Technical University of Crete (Greece)
Huiyu Zhou	Brunel University (UK)
Rodolfo Zunino	University of Genoa (Italy)
Urko Zurutuza	Mondragon University (Spain)

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)
Michal 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	Univesity 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	Univesity of Granada (Spain)
Cesar Hervas	Univesity of Cordoba (Spain)
Sebastian Ventura	Univesity of Cordoba (Spain)
Alvaro Herrero	University of Burgos (Spain)
Bruno Baruque	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 Skołud	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 Górski	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	Universidad Complutense de Madrid (Spain)
Emilio Corchado	University of Salamanca (Spain)

Organizing Committee

Robert Burduk (Chair)
Kondrad Jackowski
Bartosz Krawczyk
Maciej Krysmann
Bartosz Kurlej
Piotr Sobolewski
Szymon Sztajer
Katarzyna Tylkowska
Marcin Zmyślony
Andrzej Żołnierek

Table of Contents – Part II

Hybrid Intelligent System on Logistics and Intelligent Optimization

Metaheuristics for Combinatorial Optimization and Modelling Complex Systems

Hybrid Systems for Context-Based Information Fusion

Methods of Classifiers Fusion

Intelligent Systems for Data Mining and Applications

Systems, Man, and Cybernetics by SOCO-Workshop

Plenary

Hybrid Artificial Intelligence Systems in Management of Production Systems

Hybrid Artificial Intelligent Systems for Medical Applications

Plenary

Hybrid Intelligent Approaches in Cooperative Multi-robot Systems

Table of Contents – Part I

Plenary

General Track

Outlier Analysis for Plastic Card Fraud Detection a Hybridized and Multi-Objective Approach

Arturo Elías[1], Alberto Ochoa-Zezzatti[2], Alejandro Padilla[1], and Julio Ponce[1]

[1] Universidad Autónoma de Aguascalientes
[2] Universidad Autónoma de Ciudad Juárez
{aeliasr,apadilla,jcponce}@correo.uaa.mx, megamax8@hotmail.com

Abstract. Nowadays, plastic card fraud detection is of great importance to financial institutions. This paper presents a proposal for an automated credit card fraud detection system based on the outlier analysis technology. Previous research has established that the use of outlier analysis is one of the best techniques for the detection of fraud in general. However, to establish patterns to identify anomalies, these patterns are learned by the fraudsters and then they change the way to make de fraud. The approach applies a multi-objective model hybridized with particle swarm optimization of typical cardholder's behavior and to analyze the deviation of transactions, thus finding suspicious transactions in a non supervised scheme.

Keywords: Credit Card Fraud, Outlier Detection, Multi-Objective Optimization, Particle Swarm Optimization, Unsupervised scheme.

1 Introduction

Fraud in all its variants is an activity almost as old as mankind, which tries to take advantage of some kind, usually economic, by the fraudster with respect to shame. Specifically in the case of plastic card fraud there are several variants [15]. The total cost of plastic card fraud is high relative to other forms of payment. The first line of defense against fraud is based on preventive measures such as the Chip and PIN cards. Subsequent methods are used to identify potential fraud trying to minimize potential losses. These methods are called fraud detection systems (FDS) and usually employ a variety of proposals with the idea of detecting the majority of potential fraudulent behavior. In this model, the objective aims to identify fraud through the behavior of transactions of the cardholder. There are two major frameworks to detect fraud through statistical methods. If fraud is conducted in a known way, the pattern recognition techniques are typically used, especially supervised classification schemes [18]. On the other hand if the way in which fraud is done is not known, for example, when there are new fraudulent behaviors, outlier analysis methods are recommended [11]. Some studies show simple techniques for anomaly detection analysis to discover plastic card fraud [10]. In practical applications it is possible to use supervised and unsupervised methods together. The idea of the proposal is use time series in which transactions that have a similar behavior are grouped, so that subsequent transactions

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 1–9, 2011.

that deviate strongly from the clusters formed are candidates to be considered to have an anomalous behavior. The problem with this approach is not always abnormal behaviors are fraudulent, so a successful system must locate the true positive events, that is, transactions that are detected as fraud, but they really are fraud and not only appear to be fraudulent, where time is a factor against it, because to reduce losses, fraud detection should be done as quickly as possible. With this in mind, the proposal is strengthened by a multi-objective approach in order to improve the performance of FDS.

2 Theoretical Framework

2.1 Clusters and Outliers

The clustering is primarily a technique of unsupervised approach, although the semi-supervised clustering has also been studied frequently in recent dates [1]. Although often clustering and anomaly detection appear to be fundamentally different from one another, have developed many techniques to detect anomalies based on clustering, which can be grouped into three categories which depend on three different assumptions regarding [3]:

a) Normal data instances belong to a pooled data set, while the anomalies do not belong to any group clustered.
b) Normal instances of data are close to the cluster centroids, while anomalies are further away from these centroids.
c) The normal data belongs to large, dense clusters, whereas the anomalies belong to small and sparse clusters.

Each of the above assumptions has their own forms of detect outliers which have advantages and disadvantages between them.

2.2 Particle Swarm Optimization (PSO)

Swarm intelligence (SI) simulates the social and collective behavior of living creatures such as birds and fishes [2], but also develops global models of local interaction behavior of artificial agents. Diverse swarm intelligence algorithms have been designed and implemented, some algorithms like ant colony optimization (ACO) and particle swarm optimization (PSO) have been studied and applied in many studies and investigations [13]. In the PSO, particles fly in the course of a search space and each particle has a corresponding position and velocity at any instant of time[5].

2.3 Multi-Objective Optimization (MOO)

In real life most of the problems facing not only an objective optimization, actually there are several objectives to be achieved to have the arguments needed to make a decision. These problems must be addressed as a MOO, bearing in mind that generally, the improvement in the achievement of a goal causes deterioration of other

or others objectives. So with regard to the problems of clustering approach there are also multi-objective clustering (MOC) proposals. These methods consist in decomposing a dataset into similar groups to optimize multiple objectives in parallel [9]. Some researchers have suggested that multi-objective search and optimization could be a problem area where evolutionary algorithms (EAs) (Multi-objective Evolutionary Algorithms (MOEA)) can achieve better performance compared to other search strategies. Several MOEA techniques have developed over time and formally since 1984, however the latest trends in evolutionary strategies have been collective intelligence or swarm intelligence. Swarm intelligence has become a valuable tool to optimize operations in different businesses.

3 Research Model and Hypotheses

This approach is conducted under premise of improving the efficiency for detect fraudulent activity on plastic card transactions. In order to do this; the system is developed with a foundation of MOC, which places the problem of detecting fraud in an appropriate context to reality. In the same way, the system is strengthened through hybridization using PSO for the creation of clusters, then find the anomalies using Mahalanobis distance. The research model consists of two main dimensions: an unsupervised approach not requiring training in the absence of reliable training elements and a MOEA clustering model to identify the standard behavior in card transactions (See Figure 1).

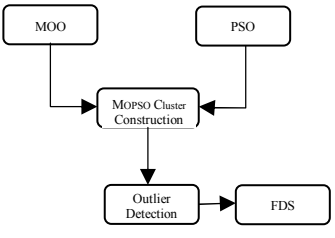


Fig. 1. Research Model

3.1 Precision

Overall accuracy is simply the percentage of correct predictions of a classifier on a test set of "ground truth". TP means the rate of predicting "true positives" (ratio of correctly predicted frauds over all of the true frauds), FP means the rate of predicting "false positives" (ratio of incorrectly predicted frauds over those test examples that were not frauds "false alarm rate"). Other two types of rates are considered for the results delivered by FDS, FN means the rate of predicting "false negatives" (ratio of no predicted frauds over all the true frauds) and TN means the rate of predicting "true negatives" (ratio of normal transactions detected). Table 1 shows the classification rate of results obtained by the FDS after analyzing a transaction.

Table 1. Classification rate of results

Outcome	Classification
Miss	False Negative (FN)
False Alarm	False Positive (FP)
Hit	True Positive (TP)
Normal	True Negative (TN)

The FDS accuracy is represented as the fraction of total transactions (both genuine and fraudulent) that are detected as correct, which can be expressed as follows [17].

$$\Pr ecision = \frac{\#\text{ of TN} + \#\text{ of TP}}{Total \text{ of carry out transaction}} . \quad (1)$$

Accuracy is the key part of the proper functioning of a FDS. We thus formally hypothesized:

H1. The level of accuracy determines the success of a FDS.

3.2 Hybridization

As in many aspects of artificial intelligence to detect abnormalities very current trend is the hybridization. The reason is because many developed algorithms do not follow entirely the concepts of a simple classical metaheuristic [12], to solve this problem is looking for the best from a combination of metaheuristics, that perform together to complement each other and produce a profitable synergy, to which is called hybridization [14]. Some possible reasons for the hybridization are: 1.- improve the performance of EAs, 2.- improve the quality of solutions obtained by EAs and 3.- incorporate EAs as part of a larger system [8]. The different instances of hybridization of metaheuristics with EAs can be grouped into different categories. The first two groups are derived from well developed taxonomy for hybridization of metaheuristics [14], which developed on the basis of their control strategies: collaborative hybrid metaheuristics and integrative hybrid metaheuristics. A third method to construct hybridization of metaheuristics with EAs is through the incorporation of the intensification and diversification (I&D) which are the two biggest issues when designing a global search method [12] where a component I&D is defined as any algorithmic or functional component having an effect of identification or diversification in the search process. Hybridization is a technique that aims to improve the performance of metaheuristics and EAs. We thus formally hypothesized:

H2. The hybridization affect positive the process of cluster construction.

3.3 Multi-Objective Pareto Front Clustering

EAs have been the most frequently used for clustering. However previous research in this respect has been limited to the single objective case: criteria based on cluster compactness have been the objectives most commonly employed, as the measures provide smooth incremental guidance in all parts of search space. In recent years there has been a growing interest in developing and applying EAs in MOO [6]. The recent studies on EAs have shown that the population-based algorithms are potential

candidate to solve MOO problems and can be efficiently used to eliminate most of the difficulties of classical single objective methods such as the sensitivity to the shape of the Pareto-optimal front and the necessity of multiple runs to find multiple Pareto-optimal solutions. In general, the goal of a MOO algorithm is not only to guide the search towards the Pareto-optimal front but also to maintain population diversity in the set of the Pareto optimal solutions[7]. In recent years, PSO has been presented as an efficient population-based heuristic technique with a flexible and well balanced mechanism to enhance and adapt the global and local exploration capabilities. The simplicity of PSO and its population based approach have made it a natural candidate to be extended for MOO [4].

H3. The multi-objective Pareto front solution using PSO affect positive the process of cluster construction.

4 Research Methodology

The FDS is running on the plastic card issuing institution. When a transaction arrived is sent to the FDS to be verified. The FDS receives the card details and purchase value to verify if the transaction is genuine, by calculating the anomalies, based on the expenditure profile of each cardholder, purchasing and billing locations, time of purchase, etc. When FDS confirms that the transaction is malicious, it activates an alarm and the financial institution decline the transaction. The cardholder concerned is contacted and alerted about the possibility that your card is at risk. To find information dynamically observation for individual transactions of the cardholder, stored transactions are subject to a clustering algorithm. In general, transactions are stored in a database of the financial institution, which contain too many attributes. In this paper we analyze three factors, the amount spent, time and location where the transaction takes place. So, if the purchase amount exceeds a certain value, the time between the uses of the card is low or the locations where different transactions are distant are facts to consider activating the alarm. All this required the calculation of anomalies through the clustering of transaction information through a multi-objective Pareto front with the support of PSO. Based on the population nature of PSO, it is desirable to produce several (different) non-dominated solutions with a single run. So, as with any other EA, the three main issues to be considered when using PSO to MOO are: (i) how to select gbest particles in order to give preference to non-dominated solutions over those that are dominated? (ii) how to retain the non-dominated solutions found during the search process in order to report solutions that are non-dominated with respect to all the past populations and not only with respect to the current one? Also it is desirable that these solutions are well spread along the Pareto front; (iii) how to maintain diversity in the swarm in order to avoid convergence to a single solution?

When solving single-objective optimization problems, the gbest that each particle uses to update its position is completely determined once a neighborhood topology is established. However in the case of MOO problems, each particle might have a set of different gbests from which just one can be selected in order to update its position. Such set of gbests is usually stored in a different place from the swarm that we will call external archive denoted as EX_ARCHIVE. This is a repository in which the

non-dominated solutions found so far are stored. The solutions contained in the external archive are used as global bests when the positions of the particles of the swarm have to be updated. Furthermore, the contents of the external archive are also usually reported as the final output of the algorithm. The following algorithm describes how a general MOPSO works.

```
Algorithm _MOPSO ( )
1. INITIALIZATION of the Swarm
2. EVALUATE the fitness of each particle of the swarm.
3. EX_ARCHIVE D SELECT the non-dominated solutions from the Swarm.
4. t = 0.
5. REPEAT
6.      FOR each particle
7.              SELECT the gbest
8.              UPDATE the Position
9.              MUTATION /* Optional */
10.             EVALUATE the Particle
11.             UPDATE the pbest
12.     END FOR
13.     UPDATE the EX_ARCHIVE with gbests.
14.     t = t + 1
15. UNTIL (t <D MAXIMUM_ITERATIONS)
16. Report Results in the EX_ARCHIVE.
```

First the swarm is initialized. Then a set of gbests is also initialized with the non-dominated particles from the swarm. As we mentioned before, the set of gbests is usually stored in an external archive, which we call EX_ARCHIVE. Later on, some sort of quality measure is calculated for all the gbests in order to select (usually) one gbest for each particle of the swarm. At each generation, for each particle, a leader is selected and the flight is performed. Most of the existing MOPSOs apply some sort of mutation operator after performing the flight. Then the particle is evaluated and its corresponding pbest is updated. A new particle replaces its pbest particle usually when this particle is dominated or if both are incomparable (i.e. they are both non-dominated with respect to each other). After all the particles have been updated, the set of gbests is updated, too. Finally, the quality measure of the set of gbests is recalculated. This process is repeated for a certain number of iterations. Once clusters are established, new transaction is entered and evaluated in the FDS, to see if it belongs to a cluster set or is outside of him, seeing the transaction as an anomaly and becoming a candidate to be fraudulent. The idea of the proposal is to work at the level of cardholder's account, keeping in main that the transaction flow of transaction logs is complex (more than 60 fields), including a unique account number. For the i_{th} account transaction has the following sequence

$$X_i = \{ x_t \,|\, x_t \in \Re^n, t = 1,2,... \}, \tag{2}$$

Where, Xi represents the i_{th} account, while x_t is the sequence of transactions for that account at time t. The Figure 2 shows the idea of the full flow of the process proposed for the FDS.

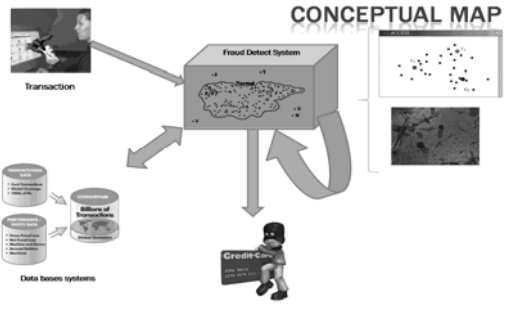


Fig. 2. Full flow of the process

As shown in the figure, the FDS is divided into two parts, one that involves the creation of clusters and the second in the detection of anomalies. Outliers detection conform the aim of this approach. Computing the outliers is a process based on Mahalanobis distance. Transactions outside of clusters are candidates to be considered fraudulent, however as mentioned above the accuracy of the system is a factor to be considered, which is expected to maximize in order to increase the functionality of the FDS. For a specific account, suppose we have legitimate transaction data, X, then our detector for new transaction x.

$$h(x|X,\gamma) = I(p(x|X,\gamma) > \theta) = \begin{cases} 1 & x \text{ is classified as a legitimate} \\ 0 & x \text{ is classified as a fraud} \end{cases} . \quad (3)$$

Here p() is a density estimate, and γ refers to control parameters for the model.

θ is the alert threshold. Difficult to set without context, but one possibility relate θ to the maximum proportion of flagged cases that we can afford to investigate.

5 Implications

Evaluate FDSs for plastic cards' using real data is too complex. Banks are generally not agree to share their data with researchers, as well as the absence of a data set for comparison (benchmark) available for experiments [16]. Therefore large-scale simulation is developed to prove the effectiveness of the system. Simulator is used to generate a mixture of genuine and fraudulent transactions. The number of fraudulent transactions in a defined amount of mixed transactions is normally distributed with mean and standard deviation specified by the user, taking the cardholder's spending behavior in his account. The mean specifies the average number of frauds in a given transaction mix. In a typical scenario, the FDS of the card issuing institution receives a large number of genuine transactions mixed moderately with fraudulent transactions, where legitimate transactions are generated from profiles of cardholders.

6 Conclusions and Future Works

It is noteworthy that achieve exact replication of the problem is not possible; although it is assumed that many of the more general features of the data used could be

reproduced in subsequent data. Fraud, however, is a known phenomenon changing in response to market conditions, as well as to measures taken by financial institutions against it, so it is quite possible that not only this but any proposal made in this regard not be accurate and universally applicable, although a fairly good approximation, which is reliable in many cases. The methodologies for the detection of fraud have their own strengths and weaknesses characteristics. The overall strength of FDS using anomaly detection is the adaptability to new patterns fraudsters, in the particular case of this study is strengthened with the application of hybridization clustering processes giving a greater dynamism to the system and making it look like a promising component within the FDS with potential advantages in regard to: upgrade and management of the heterogeneity of customers and their transactions, achieving a better accuracy in the results, and greater dynamism in the system. Additionally, the multi-objective approach place it in a better position compared to other systems, due to the characteristics of fraud detection problem where there are several factors to consider for best results.

Future work establishing the need for FDS to be increasingly proactive in order to adapt to the greatest extent possible so changing the behavior presented by fraudsters.

References

1. Basu, S., Bilenko, M., Mooney, R.: A probabilistic framework for semi-supervised clustering. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 59–68. ACM Press, Seattle (2004)
2. Beni, G., Wang, U.: Swarm intelligence in cellular robotic systems. In: NATO advanced workshop on robots and biological systems, Tuscany, Italy, pp. 26–30 (1989)
3. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys, 1–58 (2009)
4. Coello, C., Lechunga, M.: MOPSO: A proposal for multiple objective particle swarm optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation, pp. 1051–1056. IEEE Press, Hawaii (2002)
5. Cui, X., Potok, T., Palathingal, P.: Document Clustering using Particle Swarm Optimization. In: Proceedings Swarm Intelligence Symposium, 2005, IEEE, Los Alamitos (2005)
6. Deb, K.: Multi-objective optimization using evolutionary algorithms. John Wiley and Sons, Chichester (2001)
7. Dehuri, S., Cho, S.: Multi-criterion Pareto based particle swarm optimized polynomial neural network for classification: A review and state-of-the-art. Computer Science Review, 19–40 (2009)
8. Grosan, C., Abraham, A.: Hybrid evolutionary algorithms: methodologies, architectures, and reviews. In: Grosan, C., Abraham, A., Ishibuchi, H. (eds.) Hybrid evolutionary algorithms, pp. 1–17. Springer, Heidelberg (2007)
9. Jiamthapthaksin, R., Eick, C.F., Vilalta, R.: A Framework for Multi-Objective Clustering and Its Application to Co-Location Mining. In: Huang, R., Yang, Q., Pei, J., Gama, J., Meng, X., Li, X. (eds.) ADMA 2009. LNCS, vol. 5678, pp. 188–199. Springer, Heidelberg (2009)
10. Juszczak, P., Adams, N., Hand, D., Whitrow, C., Weston, D.: Off-the-peg and bespoke classifiers for fraud detection. Computational Statistics & Data Analysis (2008)

11. Kou, Y., Lu, C., Sirwongwattana, S., Huang, Y.: Survey of fraud detection techniques. In: Proccedings IEEE International Conference on Networking, Sensing and Control, pp. 749–754. IEEE press, Taipei (2004)
12. Lozano, M., García-Martínez, C.: Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. Computers and Operations Research, 481–497 (2010)
13. Özçift, A., Kaya, M., Gülten, A., Karabulut, M.: Swarm optimized organizing map (SWOM): A swarm intelligence based optimization of self-organizing map. Expert Systems with Applications: An International Journal, 640–648 (2009)
14. Raidl, G.: A unified view on hybrid metaheuristics. In: Third International Workshop, Proceedings of Hybrid Metaheuristics, pp. 1–12. Springer, Heidelberg (2006)
15. Sánchez, D., Vila, M., Cerda, L., Serrano, J.: Association rules applied to credit card fraud detection. Expert Systems with Applications: An International Journal (2009)
16. Srivastava, A., Kundu, A., Sural, S., Majumdar, A.: Credit Card Fraud Detection Using Hidden Markov Model. Transactions on Dependable and Secure Computing (2008)
17. Stolfo, S., Fan, D., Lee, W., Prodromidis, W., Chan, P.: Cost-Based Modeling for Fraud and Intrusion Detection: Results from the JAM Project. In: Proccedings DARPA Information Survivability Conference & Exposition, pp. 130–144. IEEE, Los Alamitos (2000)
18. Whitrow, C., Hand, D., Juszczak, P., Weston, D., Adams, N.: Transaction aggregation as a strategy for credit card fraud detection. Data Mining and Knowledge Discovery, 30–55 (2009)

Comparative Analysis of Recombination Operators in Genetic Algorithms for the Travelling Salesman Problem

Anca Gog and Camelia Chira

Babes-Bolyai University
1 Kogalniceanu, Cluj-Napoca
{anca,cchira}@cs.ubbcluj.ro

Abstract. The Travelling Salesman Problem (TSP) is one of the most widely studied optimization problems due to its many applications in domains such as logistics, planning, routing and scheduling. Approximation algorithms to address this NP-hard problem include genetic algorithms, ant colony systems and simulated annealing. This paper concentrates on the evolutionary approaches to TSP based on permutation encoded individuals. A comparative analysis of several recombination operators is presented based on computational experiments on a set of TSP instances. Numerical results emphasize a good performance of two proposed crossover schemes: best-worst recombination and best order recombination which take into account information from the global best and/or worst individuals besides the genetic material from parents.

Keywords: genetic algorithms, recombination, Travelling Salesman Problem.

1 Introduction

The Travelling Salesman Problem (TSP) is a well-known NP-hard problem intensively studied in operations research and computer science and commonly engaged as a standard test bed for combinatorial optimization methods. Given a number of cities and the cost of travelling (or the distance) between any two cities, TSP aims to find a minimum length closed tour that visits each city exactly once. The study of TSP is of significant importance to several application domains in planning, scheduling and logistics. TSP applications include drilling of printed circuit boards, x-ray crystallography, computer wiring, vehicle routing, order-picking problem in warehouses and scheduling problems [7].

Because TSP is NP-hard and exact solutions can not be found in polynomial time by any algorithm, there is a high interest in developing good approximation methods for solving TSP able to determine a near-optimal (or optimal) solution using reasonable resources. Heuristic approaches to TSP include genetic algorithms [6], ant colony systems [1] and simulated annealing [5].

This paper focuses on the traditional evolutionary approach to TSP by which potential solutions are represented as permutation of cities and the quality of

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 10–17, 2011.

an individual is assessed based on the corresponding tour cost. An important search operator in genetic algorithms is the recombination of two individuals which should be able to produce new potentially more efficient tours. Main existing recombination operators specific to permutation based encoding are described. Two previously introduced operators [2,3,4] - called adaptive goal guided crossover (AGGX) and best-worst crossover (BWX) - are presented. Furthermore, we introduce the best order crossover (BOX) operator. AGGX, BWX and BOX rely on different schemes for the recombination of genetic material from parents, global best individual, global worst individual and/or the parent's line best anccestor individual. The comparative performance of all these recombination operators inside a standard genetic algorithm is analysed. The study is based on extensive numerical experiments for various instances from the TSP library [8].

2 The Travelling Salesman Problem

A set of k points in a plane is given, corresponding to the location of k cities. The Travelling Salesman Problem requires finding the shortest closed path that visits each city exactly once. The problem can be formalized as follows:

A set of k cities

$$C = \{c_1, c_2, \ldots, c_k\}$$

is given. For each pair

$$(c_i, c_j), i \neq j,$$

let

$$d(c_i, c_j)$$

be the distance between the city c_i and the city c_j. One has to find a permutation π' of the cities

$$(c_{\pi'(1)}, \ldots, c_{\pi'(k)}),$$

such that

$$\sum_{i=1}^{k} d(c_{\pi'(i)}, c_{\pi'(i+1)}) \leq \sum_{i=1}^{k} d(c_{\pi(i)}, c_{\pi(i+1)}),$$

$$\forall \pi \neq \pi', (k+1 \equiv 1).$$

TSP can be defined as the search for a minimal Hamiltonian cycle in a complete graph.

The simplest evolutionary approach to this NP-hard problem is outlined in what follows. A potential solution for the problem (a chromosome) is a string of length k that contains a permutation π of the set

$$\{1, \ldots, k\},$$

and represents the order of visiting the k cities. Let S denote the search space (the permutation set). Fitness assignment is applied by the criterion function f given by:

$$f : S \to \Re^+, f(\pi) = \sum_{i=1}^{k} d(c_{\pi(i)}, c_{\pi(i+1)}),$$

$$(k+1 \equiv 1),$$

The fitness function is to be minimized. Thus, the fitness of a chromosome is the length of the Hamiltonian path that visits the cities in the order specified by the permutation.

3 Crossover Operators for Permutation Based Encoding

A brief review of recombination operators for permutation based encoding [6] is presented in what follows. All these crossover operators are compared via numerical experiments with the recombination operators proposed in the next section.

The *Order Crossover (OX)* is based on the idea that the order of alleles and not their positions are relevant. The offspring are obtained by keeping a sequence of alleles from one parent and the other alleles from the second parent - but in the same order. In the *Order Based Crossover (OBX)* several alleles from one parent are selected and their order in the offspring is imposed by the other parent. In the *Edge Recombination Crossover (ERX)*, the edges of a tour are considered as containing the relevant genetic material. It actually considers that the path representation does not contain enough information and therefore, the edges list is added. When using the *Partially-Mapped Crossover (PMX)*, a sequence of one parent is mapped onto a sequence of the other parent and the rest of the alleles are exchanged. In *Cycle Crossover (CX)* the offspring is created by filling each position with the corresponding allele from one of the two parents. *Maximal Preservative Crossover (MPX)* is similar to PMX, but it has the advantage that it destroys a limited number of edges. When using the *Alternating-Position Crossover (APX)*, the offspring is created by taking one allele from each parent's first position, continuing with the second position and so on, ignoring the alleles already existing. *Position Based Crossover (PBX)* selects several alleles from one parent and their positions is imposed on the corresponding alleles of the second parent.

4 Proposed Recombination Operators

4.1 Adaptive Goal Guided Recombination

The *Adaptive Goal Guided Recombination (AGGX)*, originally introduced in [2] and further improved in [3], is passing to the offspring not only genetic information from the parents, as a standard crossover, but certain genetic information from the parents best ancestors (*LineBest*) and from the best global (*GlobalBest*) as well.

The concept of relevant genetic material is introduced in this context as representing the chromosomal information that can be retrieved in both *LineBest* and *GlobalBest*. The fact that genetic information from the *GlobalBest* has been preserved in individuals that represent the best ancestors of some chromosomes from the current population, allows us to consider it as being relevant for the search process. We use this information in order to accelerate the search process by orienting it towards promising regions of the search space.

In order to be able to differentiate between good and bad genetic material, we have to decide what relevant genetic material means when considering permutation based encoding. In TSP for example, it is not important that a certain city has been visited at a certain moment of time, but rather the succession of visited cities, because the edges of a tour can be seen as the carriers of the genetic information, according to [9].

Because preserving diversity in the search space, particularly in the first stages of an evolutionary algorithm, represents a condition for avoiding the search to become trapped into a local optimum, the amount of relevant genetic information passed to the offspring is controlled by taking into account the number of the current generation related to the total number of generations.

Let us consider *NoEdges* to be the total number of common genes of *GlobalBest* and *LineBest*, *NoCrt* the number of the current generation and *NoGen* the number of generations after which the algorithm ends if no better solutions are found. The number of (randomly chosen) genes kept in the offspring is given by:

$$NoKept = NoEdges * e^{-\frac{m*NoGen-NoCrt}{m*NoGen}},$$

where m is a factor that controls the length of the sequences of genes passed to the offspring. Experimental results have shown that 10 is a good value for this parameter.

NoKept is an exponentially increasing function that ensures that the diversity is increased in the first stages of the algorithm and the search becomes more goal-oriented in the final stages by keeping in the configuration of the offspring more relevant genetic information from the *LineBest* and the *GlobalBest*. We also transfer to the offspring a randomly chosen sequence of genes from one parent.

4.2 Best-Worst Recombination

The *Best-Worst Recombination (BWX)* scheme, introduced in [4], is performed in an environment where each individual has extra knowledge about the best individual (*GlobalBest*) obtained so far in the search process and also about the worst individual *GlobalWorst* obtained so far. The goal of BWX is twofold: to use the good genetic material contained in the *GlobalBest* while performing recombination and to avoid transmitting to the offspring bad genetic material already contained in the *GlobalWorst* during the search process.

If one of the parents contains genetic material that can be also retrieved in the *GlobalBest*, we consider that these genetic traits should be also passed to

the offspring in order to accelerate the search process. This strategy will save time needed to find good characteristics for the offspring, by using the already obtained ones. An exploitation of the search space is achieved by means of this strategy, because we choose good genes that should be passed to the offspring and we realize a local search around them.

If one of the parents contains genetic material that can be also retrieved in the *GlobalWorst*, we pass to the offspring a permutation of that succession of edges. This strategy will save search time by avoiding those configurations that prove themselves to represent bad genetic material, as part of the worst individual obtained so far. The exploration of the search space is also performed by the perturbation of these configurations. However, not all genetic material contained by the *GlobalWorst* can be considered as being not good to the search process: there might be sequences of genes that represent good genetic material, but combined with other sequences of genes lead to the worst configuration obtained so far. In order to avoid the extinction of good genetic material, when performing the permutation of the sequences that belong to both parent and *GlobalWorst*, the probability that we obtain the same sequence is equal to the probability of obtaining any other permutation of those genes. For the same reason, if there are sequences belonging to both *GlobalBest* and *GlobalWorst*, these will be considered as good genetic material and passed to the offspring in that exact form.

4.3 Best Order Crossover

The new proposed *Best Order Crossover (BOX)* also exploits the fact that the order of the cities is important, not their positions. The main new feature of the proposed crossover operator is the use of genetic material belonging to the *GlobalBest* individual together with genetic information from the two parents that are subject to recombination.

Several cutting points are randomly chosen. The number of cutting points is randomly selected and can be even zero. Every two consecutive cutting points (including the beginning and the end of the chromosome array) will generate a sequence of alleles; when the number of cutting points is 0, we will only have one sequence containing the whole chromosome.

One of the following values will be assigned to each resulting sequence: -1, -2, -3. These values identify the source used for creating the offspring. A sequence identified by -1 means that the alleles will be taken from the main parent. A sequence identified by -2 means that the alleles will be taken from the other parent and when -3 is assigned to a sequence, the alleles will be taken from *GlobalBest*.

For example, in order to create the first offspring, we consider the first parent as the main parent. When we have a -1 sequence in offspring, we take the corresponding positions in the same order from the main parent. For a -2 sequence, we take the corresponding positions from the main parent but the order is given by the other parent. For a sequence identified by -3, we take the corresponding positions from the main parent but in the order imposed by *GlobalBest*.

5 Experimental Results

Several TSP instances [8] are considered in order to provide a comparative analysis of the recombination operators for permutation based encoding.

A standard Genetic Algorithm (GA) is considered for numerical experiments. The population consists of 100 individuals and roulette selection, inverse mutation with 0.05 mutation rate and the various recombination operators are used. Furthermore, elitism ensures that the fitness of the best solution in the population does not deteriorate as the generation advances. For the considered TSP instances, the GA is applied with the most popular recombination operators for permutation based encoding [6] described in Section 3 and the proposed AGGX, BWX and BOX operators presented in Section 4.

Table 1 presents the average results over 10 runs obtained after 1000 generations. The columns represent the TSP instances used in experiments while the lines of the table give the GA results based on 11 different recombination operators.

The test results indicate the acceleration of the search process when using the AGGX, BWX and BOX recombination operators (last three lines in Table 1). Moreover, the introduced BOX outperforms all the other operators for all considered problems.

Table 1. Average GA results over 10 runs obtained after 1000 generations with all considered recombination operators for 10 TSP instances (given in columns)

	EIL51	ST70	PR76	EIL76	KROA100	LIN105	PR124	TS225	GIL262	PR299
OX	486	873	153359	708	36992	28907	149098	509356	9322	247282
OBX	623	1424	238253	1082	70942	51956	314630	980075	16847	479800
ERX	594	1383	208515	1060	73739	43120	233143	873696	16703	414322
PMX	520	1010	162829	751	44563	32064	173774	583968	10487	307267
CX	593	1247	205497	969	62446	41386	239291	757458	13286	344999
MPX	841	1739	284426	1265	83980	55282	317210	811696	14387	357980
APX	1067	2394	407957	1826	122240	88653	498100	1336437	22599	626387
PBX	593	1429	215691	983	72991	52944	311195	1103359	19434	513789
AGGX	597	945	206457	856	34876	25763	135974	537425	12876	354765
BWX	464	792	141767	652	33652	23342	129811	484764	9208	226592
BOX	**460**	**741**	**121591**	**605**	**27563**	**19575**	**93154**	**383227**	**6937**	**172867**

The performance of BWX is clearly better than that of AGGX emphasizing the advantages of considering the information generated by the *GlobalWorst* solution in addition to the genetic material from the parents and the *GlobalBest*. It is interesting to observe that BOX performs better than AGGX and BWX although it relies on a more simple scheme. BOX takes into account positions order from *GlobalBest* only compared to AGGX which also considers *LineBest* or to BWX which also considers *GlobalWorst*.

Figure 1 depicts the improvement in percentages generated by AGGX, BWX and BOX after 1000 generations compared to the best solution reported by all other recombination operators.

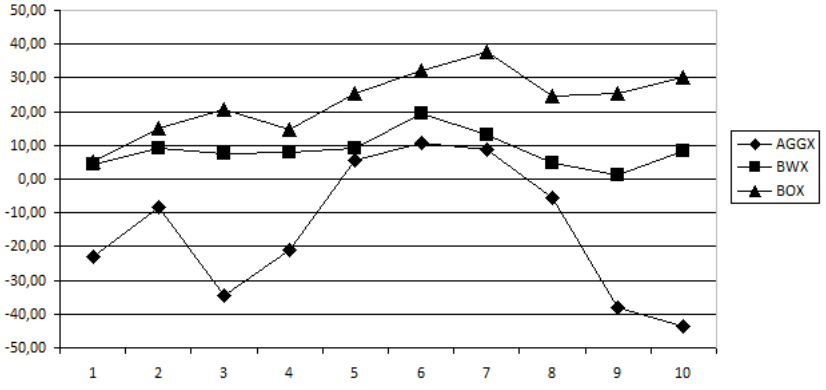


Fig. 1. Solution improvement of AGGX, BWX and BOX after 1000 generations

The analysis of solution improvement confirms BOX as the most efficient operator. AGGX was able to bring an improvement compared to other operators only for a few TSP instances while BWX is able to steadily improve solution quality but a lower rate compared to BOX.

6 Conclusions

Several recombination operators for permutation based encoding in GAs are analysed and compared for a set of TSP instances. Numerical results indicate a superior performance of the BWX and BOX recombination operators able to bring a significant improvement in the solution quality compared to both the best and average results from the other considered operators. The main features which make BWX and BOX more efficient refer to combining genetic material from the selected parents while considering the best (and/or worst) potential solutions obtained up to the current generation in the search process. Comparing BWX to BOX, we notice an interesting behaviour as BOX is capable to obtain overall better results based on a more simple recombination scheme compared to BWX. Further numerical experiments will be carried out to extend the current results to other TSP instances and problems that can be addressed with a permutation-based encoding in a genetic algorithm.

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. Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation 1, 53–66 (1997)

2. Gog, A., Dumitrescu, D.: Adaptive Search in Evolutionary Combinatorial Optimization. In: Proceedings of the International Conference Bio-Inspired Computing – Theory and Applications (BIC-TA 2006), Wuhan, China, pp. 123–130 (2006)
3. Gog, A., Dumitrescu, D.: A New Recombination Operator for Permutation Based Encoding. In: Proceedings of the 2nd International Conference on Intelligent Computer Communication and Processing (ICCP 2006), pp. 11–16 (2006)
4. 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)
5. van Laarhoven, P.J.M., Aarts, E.H.L.: Simulated Annealing: Theory and Applications. Kluwer Academic Publishers, Dordrecht (1987)
6. Larranaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., Dizdarevic, S.: Genetic Algorithms for the Traveling Salesman Problem: A Review of Representation and Operators. Articial Intelligence Review 13, 129–170 (1999)
7. Matai, R., Singh, S., Mittal, M.L.: Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches. In: Davendra, D. (ed.) Traveling Salesman Problem, Theory and Applications, InTech (2010) ISBN: 978-953-307-426-9
8. Reinelt, G.: TSPLIB - A Traveling Salesman Problem Library. ORSA Journal of Computing, 376–384 (1991)
9. Wagner, S., Affenzeller, M., Schragl, D.: Traps and Dangers when Modelling Problems for Genetic Algorithms, Cybernetics and Systems. Austrian Society for Cybernetic Studies, 79–84 (2004)

Multiple Local Searches to Balance Intensification and Diversification in a Memetic Algorithm for the Linear Ordering Problem

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

Instituto Tecnológico de Ciudad Madero, México
1o. de Mayo y Sor Juana I. de la Cruz S/N C.P. 89440
Cd. Madero, Tamaulipas, México
hfraire@prodigy.net.mx, gpe_cas@yahoo.com.mx, cggs71@hotmail.com, jjgonzalezbarbosa@yahoo.com.mx, r_pazos_r@yahoo.com.mx, b_shulamith@hotmail.com, david_teran01@yahoo.com.mx

Abstract. The Linear Ordering problem (LOP) is an NP-hard problem, which has been solved using different metaheuristic approaches. The best solution for this problem is a memetic algorithm, which uses the traditional approach of hybridizing a genetic algorithm with a single local search; on the contrary, in this paper we present a memetic solution hybridized with multiple local searches through all the memetic process. Experimental results show that using the best combination of local searches, instead of a single local search, the performance for XLOLIB instances is improved by 11.46% in terms of quality of the solution. For the UB-I instances, the proposed algorithm obtained a 0.12% average deviation from the best known solutions, achieving 17 new best known solutions. A Wilcoxon test was performed, ranking the proposed memetic algorithm as the second best solution of the state of the art for LOP. The results show that the multiple local searches approach can be more effective to get a better control in balancing intensification/diversification than the single local search approach.

Keywords: Metaheuristics, Memetic algorithm, Linear Ordering Problem, Local Search, Intensification/diversification balance.

1 Introduction

Given a C_{ij} matrix of weights of size $n \times n$, the linear ordering problem (LOP) consists in finding a permutation p of columns (and rows) such that the sum of the weights in the upper triangle is maximized. Its mathematical expression is:

$$\max\left(C_{LOP}(p) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{p_{(i)}\ p_{(j)}} \right) \tag{1}$$

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 18–25, 2011.

Here p_i is the index of column (and row) i in the permutation. In LOP, the permutation p provides the ordering of rows and columns, as described by Laguna [1].

The Linear Ordering Problem has been described as an NP-hard problem by Karp and Thatcher [2] and Garey and Johnson [3].

Important applications of LOP exist in scheduling, social sciences, electronics, archeology, and particularly in economics, where it is applied to solve the problem of triangulating the input-output tables in the economic model by Leontief [4].

2 Related Work

The memetic algorithm by Schiavinotto and Stutzle [5] currently offers the best solution of the state of the art for LOP. In this work a genetic algorithm was hybridized with a single local search on an insertion neighborhood. The single local search included in the memetic algorithm uses a criterion between first and best. The final configuration of memetic includes a population size of 25 individuals, an offspring formed by 11 children generated by OB crossover, a diversification by reinitializing of population (after 30 iterations have occurred without changes in fitness average). The final memetic algorithm didn´t include a mutation operator.

One of the best solutions for LOP is the Tabu search by Laguna et al. [1]. This solution includes an intensification phase using short-term memory based on a tabu criterion, a diversification process through a long-term memory that uses a frequency register, and an additional intensification process that applies a path relinking strategy based on elite solutions. The Tabu search algorithm uses two different local searches (first and best) in its different processes. These local searches explore an insertion neighborhood through consecutive swap movements.

A benchmark library and a comprehensive study of heuristic methods were developed by Martí [6]. In this work, under the same experimental conditions and on the same standard sets of instances (OPT-I and UB_I), 10 of the top performing heuristic algorithms are assesed. The highest performing algorithms were the memetic algorithm and Tabu search.

Memetic algorithms are commonly implemented as evolutionary algorithms endowed with a single local search component [7]. The use of multiple local searches in the memetic algorithm is associated to learning approaches [8], but this strategy has a high computational cost. On the contrary, we propose a less expensive new approach by incorporating in a memetic algorithm multiple local searches with different levels of intensification, according to the requirements of the processes included in the algorithm.

3 Local Searches

The insertion neighborhood used by the local searches implemented, is defined by the insert operation: $\pi \times \{1, \ldots, n\}^2 \rightarrow \pi$.

$$insert(\pi,i,j) \triangleq \begin{cases} (\ldots,\pi_{i-1},\pi_{i+1},\ldots,\pi_{j},\pi_{i},\pi_{j+1},\ldots) & i < j \\ (\ldots,\pi_{j-1},\pi_{i},\pi_{j},\ldots,\pi_{i-1},\pi_{i+1},\ldots) & i > j \end{cases} \quad (2)$$

This neighborhood was implemented applying the exploration strategy inspired on the Dynasearch method proposed by Congram [9]. The insertion is made by a series of swap movements between adjacent elements, where $i = j \pm 1$. The change in the objective function value is given by:

$$\Delta_s(\pi,i,j) \triangleq \begin{cases} c_{\pi_i\pi_j} - c_{\pi_j\pi_i} & i = j+1; \\ c_{\pi_j\pi_i} - c_{\pi_i\pi_j} & i = j-1. \end{cases} \quad (3)$$

Since the change evaluation for each movement is constant, the cost to evaluate the neighborhood is $O(n^2)$. For optimizing the execution time, the costs of all the swap moves are pre-calculated like in the Tabu search by Laguna [1]. The expression for this calculation is:

$$d_{ij} = c_{ij} - c_{ji} \quad \forall i,j = 0..\ n\text{-}1. \quad (4)$$

In this work two local search algorithms with different levels of intensification/diversification were implemented. The algorithm in Figure 3 corresponds to LS_f local search [5], which has an external cycle to examine all the sectors in the permutation order. For each sector, an internal cycle evaluates all the neighbors searching for the best (if none is better it returns the less worse neighbor solution).

```
LS_f Algorithm(π)
  for (i = 0.. n-1) do
    r⁻ ← arg max r. r = i f (insert(π,i,r⁻))
    π' = insert ((π,i,r⁻))
    if f (π') > f (π) then
      return (π')
    end_if
  end_for
  return (π)
end_LS_f Algorithm (π)
```

Fig. 1. LS_f algorithm

On the other hand the *Best* local search [1] is much more intensive: it starts from a feasible solution which is improved over the time. All sectors are scanned in the permutation order; all the consecutive positions forward and backward from the current position are examined, choosing the position which produces the highest increment (or the lowest decrease) in the objective function value for carrying out the movement. The process is repeated as long as the current solution is improved, Figure 4 shows the algorithm for the *Best* local search. It is worth mentioning that the *Best* local search is more intensive and its cost is greater than the cost of the LS_f algorithm. As we can see, both searches have a variable cost.

```
Best Algoritm (π)
        do
                for (i = 1.. nsectors)  hacer
                        r⁻ ←arg max r. r≠i f (insert(π,i,r⁻ ))
                        π'= insert ((π,i,r⁻))
                end_for
        while ((f(π') > f(π))
        return (π')
end Best Algoritm (π)
```

Fig. 2. *Best* algorithm

4 Proposed Memetic Algorithm (MLSM)

The proposed memetic algorithm that we denote by the acronym MLSM (multiple local search memetic) is hybridized with different local searches. Figure 1 shows the pseudo-code for MLSM.

```
MLSM Memetic Algorithm
        Population←RandomGeneratepopulation( );
        ImprovePopulation ←BestLocal Search(Population);
        while (not stop condition)
                if stagnation  then
                        Population←Selectbestindividual(Population);
                        Population← RandomGeneratepopulation( );//diversification
                        ImprovePopulation← LSFLocalSearch(Population);
                end_if
                for i←1..#crossovers do
                        select π_a, π_b from Population( );
                        offsprings←OB_crossover( );
                        ImprovePopulation← BestLocalSearch(Population);
                end_for
                Population←SortPopulation(Population);
                best_individual←Select_best(SortedPopulation)
        end_while (reach stop condition);
end_Memetic Algorithm
```

Fig. 3. Proposed memetic algorithm MLSM

In the proposed memetic algorithm, the initial population is obtained by generating randomly 36 individuals, to which the *Best* local search is applied.

For subsequent generations, two parents are randomly selected for applying crossover. From the first parent a set of positions are randomly selected (0.4 * instance size), then the values in the non-selected positions are copied directly to the corresponding positions of the offspring. In the next step, the values in the selected positions are ranked according to the order in the second parent and copied to the vacant positions of child. Figure 2 shows this crossover process. In the selection

process, the first father is randomly chosen according to a uniform distribution and the second parent is chosen giving preference to the best elements of the population. A general condition for the selection of parents is that they haven't recently generated offspring. The *Best* local search is applied to the new population.

To build the new population, the 25 best individuals are chosen from the current population and from the generated offspring, duplicates are eliminated.

An extra diversification by randomly regenerating the individuals in the current population is applied, remaining only the best individual. This process is triggered to avoid premature convergence when the average of the population objective function does not change during five consecutively generations. In this case an LS_f Local search is applied to all the individuals in the new population.

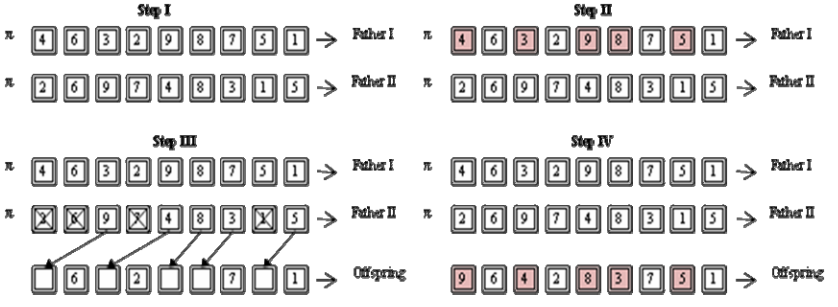


Fig. 4. An OB crossover example

5 Experimental Results

Experimental work was conducted on a Dell Power Edge 2600 with a XEON processor at 3.06 GHZ and 4GB in RAM. The algorithm was coded in C++, and compiled in Visual Studio 2005. Each algorithm was executed using two time limits (10 and 600 seconds) for each instance and a seed of 1471. The instances were grouped into four sets: RandAII, Spec, RandB, XLOLIB, and RandAI.

A first experiment to assess different combinations of the local search strategies (*Best* (B) and LS_f (LSF)) on the memetic algorithm was performed. They are applied to the initial population, to the population after the crossover and to the population after the diversification process. The experiment was carried out using the XLOLIB instances. Table 1 shows the results for the best combinations. The first column shows the combination of local searches. The combination (B, B, B) corresponds to a single local search memetic algorithm (SLSM, traditional approach), that is used as base case. The third column includes the average deviation from the best known solution (%), and the fourth column contains the difference in performance for the corresponding combination with respect to the performance of the base combination. The best combination (B, LSF, B), highlighted in the table, achieves a performance improvement of 11.4% with respect to the base case. In the table, the acronym NA indicates that no local search was applied.

A second experiment was carried out to assess the performance of the proposed memetic algorithm, which incorporates the best configuration of local searches. The MLSM algorithm was compared average wise with the two best metaheuristics: tabu search by Laguna (TS) [1] and memetic algorithm by Schiavinotto (SM) [5]. The results for these two metaheuristics were obtained from the benchmark library developed by Marti et al. [6] and are shown on Table 2.

Table 1. Average deviation from best known solutions and percentage improvement with respect to the base SLSM algorithm

Combination of local searches	Memetic algorithm	Average deviation from best known (%)	Percentage improvement versus base SLSM
B, B, B	Base SLSM	0.18974923	0
LSF, LSF, B	MLSM	0.19692336	-3.78084802
B, LSF, B	MLSM	0.16805435	11.43344824
LSF, B, B	MLSM	0.19661945	-3.62068399
NA, NA, B	SLSM	0.17349502	8.566153338
LSF, LSF, LSF	SLSM	0.42988623	-126.554927
B, LSF, LSF	MLSM	0.40227557	-112.003796

Table 2. Experimental results for UB-I instances (10 sec.)

Instances	Performance	TS by Laguna [3]	SM by Schiavinotto [5]	MLSM
RandA1	% Error (Avg)	0.12	0.05	0.11
	# Best	5	33	19
RandA2	% Error (Avg)	0.01	0	0.001
	# Best	3	39	33
RandB	% Error (Avg)	0	0	0.27
	# Best	20	20	0
XLOLIB	% Error (Avg)	0.61	0.12	0.169
	# Best	0	2	1
Spec	% Error (Avg)	0.45	0.049	0.073
	# Best	3	3	4
% Average Error		0.23	0.04	0.12
#Total Best		31	97	57

As can be seen, the proposed MLSM algorithm achieves a comparable performance to the memetic algorithm by Schiavinotto (SM). Afterwards non-parametric Wilcoxon tests were performed to compare MLSM with respect to the tabu and memetic algorithms, applying time limits of 10 and 600 seconds. MLSM significantly outperforms the Tabu solution for the two different time limits (the results are not shown by space restrictions). The results of the experiment with SM algorithm are shown in the Table 3. In this table, when R^+ is larger than R^-, the SM algorithm is better than MLSM, the opposite is true if R^- is greater than R^+. If the greatest of R^+ and R^- is greater than the reference value (R), then the performance difference is statistically significant, otherwise both algorithms have the same

performance. In all cases, considered reliability is 0.05. As we can see, the memetic algorithm by Schiavinotto[5] outperforms significantly the performance of MLSM in 6 of 10 cases for 10 seconds. In the test with 600 seconds MLSM was outperformed by SM only in 3 of 10 cases. The performance increment of the MLSM algorithm could be explained because the multiple searches approach permitted to incorporate a better balance of intensification/diversification than the single search approach.

Additionally, with the proposed memetic algorithm MLSM 17 best known solutions were improved. These new best known solutions were obtained when the MLSM algorithm was carried out using a time limit of 600 seconds. Table 4 shows these new best known solutions.

Table 3. The Wilcoxon test (memetic algorithm by Schiavinotto [5] and MLSM)

Execution time limit of 10 seconds										
Prom.	RandA I(100)	RandA I(150)	RandA I (200)	RandA I(500)	RandA II(150)	RandA II(200)	RandB	XLOLIB (150)	XLOLIB (250)	Spec
N	25	25	25	25	25	25	20	39	39	7
N	7	24	25	25	4	12	20	39	33	3
R^+	23	282	246	325	10	55	210	749	558	4
R^-	5	18	79	0	0	23	0	31	3	2
R	26	219	236	236	NA	65	158	531	391	NA
Winner	none	SM	SM	SM	NA	none	SM	SM	SM	NA
Dif.Sig.?	No	Yes	Yes	Yes	NA	No	Yes	Yes	Yes	NA
Execution time limit of 600 seconds										
N	25	25	25	25	25	25	20	39	39	7
N	0	8	16	15	0	3	0	39	1	3
R^+	0	36	128	88	0	6	0	537	1	4
R^-	0	0	8	32	0	0	0	243	0	2
Ref	NA	33	107	95	NA	NA	NA	531	NA	NA
Winner	SM=MLSM	SM	SM	SM=MLSM	SM=MLSM	NA	S=MLSM	S	S=MLSM	NA
Dif.Sig.?	No	Yes	Yes	No	No	NA	No	Yes	No	NA

Table 4. New best solutions

Set	Instances	Old *best solutions*	New *best solutions*
Rand AI	N-t1d200.13	409234	409270
	N-t1d500.15	2411718	2412400
	N-t1d500.16	2416067	2416446
	N-t1d500.17	2401800	2402438
	N-t1d500.18	2421159	2421511
Spec	N-atp134	1796	1797
	N-atp163	2073	2075
XLOLIB	N- t75d11xx_150.mat	9642140	9643446
	N - tiw56r67_150.mat	2056347	2056446
	N - stabu2_250.mat	11500448	11500845
	N - stabu3_250.mat	11900315	11901939
	N - t59d11xx_250.mat	3841167	3841376
	N - t75d11xx_250.mat	25017059	25022475
	N - t75n11xx_250.mat	4524942	4525197
	N - tiw56n54_250.mat	2098726	2098877
	N - tiw56n62_250.mat	4142745	4143351
	N - tiw56n72_250.mat	11149706	11151094

6 Conclusions

In this work the linear ordering problem is approached. We propose a memetic algorithm that incorporates multiple local searches instead of a single local search. A combination of different local searches is evaluated for the process of improving the individuals.

The MLSM algorithm was tested on the UB-I instances, obtaining 0.12% in the average deviation from the best known solution, comparable with the memetic algorithm by Schiavinotto, and it outperforms the Tabu by Laguna by 48% in solution quality. The Wilcoxon non-parametric test shows that MLSM significantly outperforms the Tabu solution; but the memetic algorithm by Schiavinotto outperforms significantly the performance of MLSM in 6 of 10 cases for 10 seconds. In the test with 600 seconds MLSM was outperformed by SLSM only in 3 of 10 cases. The recovery of the MLSM performance could be explained because the multiple local searches approach permits to get a better control of the balance of intensification/diversification than the single local search approach. With the proposed memetic algorithm MLSM, 17 best known solutions were improved. We are currently applying the multiple local searches approach in other metaheuristics.

Acknowledgements. This work was supported by PROMEP, CONACYT and COTACYT. Also we want to thank Manuel Laguna and Abraham Duarte, for their support.

References

1. Laguna, M., Martí, R., Campos, V.: Intensification and diversification with elite tabu search solutions for the linear ordering problem. Computers and Operations Research 26, 1217–1230 (1998)
2. Karp, R., Miller, R., Thatcher, J.: Reducibility among Combinatorial Problems. In: Complexity of Computer Computation, pp. 85–103.Plenum Press, New York (1972)
3. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Co, New York (1975)
4. Leontief, W.W.: Input-Output Economics. Oxford University Press, Oxford (1986)
5. Schiavinotto, T., Stützle, T.: Search Space Analysis of the Linear Ordering Problem. Darmstadt University of Technology, Intellectics Group, Darmstadt (2004)
6. Martí, R., Reinelt, G., Duarte, A.: A Benchmark Library and a Comparison of Heuristic Methods for the Linear Ordering Problem. Computational optimization and applications (2010)
7. Moscato, P., Cota, C.: A Modern Introduction to Memetic Algorithms. In: Gendreau, M., Potvin, J.-Y. (eds.) International Series in Operations Research & Management Science, pp. 449–468. Springer, Heidelberg (2010)
8. Ong, Y.-S., Keane, A.J.: Meta-Lamarckian Learning in Memetic Algorithms. IEEE Trans. Evol. Comput. 8(2), 99–110 (2004)
9. Congram Richard, K.: Polynomially Searchable Exponential Neighborhoods for Sequencing Problems in Combinatorial Optimization. Faculty of Mathematical Studies, University of Southampton, UK (2000)

Enhancing Accuracy of Hybrid Packing Systems through General-Purpose Characterization

Laura Cruz-Reyes[1], Claudia Gómez-Santillán[1], Satu Elisa Schaeffer[2], Marcela Quiroz-Castellanos[1], Victor M. Alvarez-Hernández[1], and Verónica Pérez-Rosas[1]

[1] Instituto Tecnológico de Ciudad Madero, ITCM
[2] Facultad de Ingeniería Mecánica y Eléctrica, UANL
lcruzreyes@prodigy.net.mx, cggs71@hotmail.com, elisa@yalma.fime.uanl.mx, qc.marcela@gmail.com, {mantorvicuel,x_filesrecords}@hotmail.com

Abstract. Some Hybrid Packing Systems integrate several algorithms to solve the bin packing problem (BPP) based on their past performance and the problem characterization. These systems relate BPP characteristics with the performance of the set of solution algorithms and allow us to estimate which algorithm is to yield the best performance for a previously unseen instance. The present paper focuses on the characterization of NP-hard problems. In related work, characterization metrics are traditionally oriented towards problem structure. In this work, we propose metrics based on descriptive statistics for the Bin Packing Problem (BPP). The proposed metrics are of general purpose, meaning that the metrics do not depend on problem structure and can be applied to BPP and other problems to complement existent metrics. The "enhanced" Hybrid Packing System outperforms the version that does not take advantage of the general-purpose metrics; the results obtained show a 3%-improvement with respect to the reference Packing System.

Keywords: Hybrid Solution Systems, Bin Packing Problem, Problem Characterization, Heuristic Algorithms, Algorithm Selection.

1 Introduction

The benefits and shortcomings of *approximate solution algorithms* for complex optimization problems are widely studied. However, the lack of formal methods to characterize the performance of such algorithms makes it difficult to evaluate and select among them [1, 2, 3, 4].

In this work we seek to characterize NP-hard optimization problems in order to select the best approximation algorithm for the solution of a given problem instance; the algorithm portfolio is part of a hybrid solution system. As a practical example, we attend the *BPP*. The proposed characterization process is based on *statistical techniques* and *machine learning*, and forms part of a methodology oriented to the construction of prediction models of algorithmic performance [5]. A prediction model is constructed based on a learning set of problem instances and is then applied to infer the best-performing algorithm for new, previously unseen problem instances.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 26–33, 2011.

The set of learning instances is the outcome of a characterization process for optimization problems and class-transformation techniques. For each instance, the characterization process extracts difficulty metrics, and identifies - among a set of available algorithms the one that yields the best performance. The class-transformation methods conserve for each instance only the best-performing algorithm in the set. The transformation is necessary in setups where it is feasible to execute only one solution algorithm per each problem instance.

For characterization, five metrics specific to the problem were used and 21 new metrics based on classical statistical descriptive measures were incorporated. A set of standard BPP instances was used to validate the performance of the prediction model. Experimental results show a prediction accuracy of 79% surpassing the 76% previously reported [1]. However, these percentages represent an upper bound from exact result of prediction model accuracy for which reason it was necessary to establish a fair measure.

2 Algorithm Selection for a Hybrid Solution System

The present work extends a Methodology for Development of Hybrid Solution Systems based on Algorithms Selection [5]. Using this methodology, critical characteristics for a solution instance are identified and the performance of available solution algorithms is evaluated. Through a learning process, a prediction model that relates the characteristics of the instance to the performance is built. Such models make it possible for a new instance to infer the algorithm with the best performance. This system is composed of three main elements: problem and algorithm characterization, building of a prediction model, and the best-performing algorithm inference. The extension takes advantage of three proposed strategies: general-purpose characterization metrics, multiclass instances and realistic model evaluation.

3 Characterization of Optimization Problems

The characterization process for an instance *I* of an optimization problem consists in transforming its parameters into a set of metrics, which aim to represent the solution difficulty. In algorithm selection, these metrics are used to solve an instance *I*, inferring based on the metrics which algorithm will respond with the best performance. Each metric must be represented with a unique numeric value, and preferably mutually independent of the other metrics. Also, the time necessary to calculate the metric set must be relatively small in comparison to the time required for the inference and the execution of the selected algorithm [6].

Despite of its practical and theoretical importance, little effort has been dedicated investigating problem structure in previous work. This may be partially due to the difficulty in characterizing the structure of complex real-world problems [3, 4, 7].

Previous works propose metrics that quantify characteristics specific to problem structure [5, 6, 8]. Some propose the use of metrics based on statistical measures [9]. In this work, measures of both types are formulated for the Bin Packing Problem. Also, the necessity of instance transformation methods is identified for instances that

have more than one algorithm associated with the best performance (solution quality and execution time). The nature of the algorithm selection problem requires that the prediction model infers a unique algorithm for each given instance; that is because we consider it feasible to execute only a single algorithm.

4 Study Case: BPP Solved with Heuristic Algorithms

4.1 Bin Packing Problem

This work considers a discrete one-dimensional version of BPP, this is a classic NP-Hard combinatory optimization problem, in which there is a sequence n objects, that will be distributed of $L = \{a_1,a_2,...,a_n\}$, each object with a size $0 < s(a_i) \leq c$, and unlimited bins, where each of them have c capacity. The aim consists in finding the distribution of objects inside of a set of identical bins. Objects are allocated inside the bins in such a way that the bin capacity is not exceeded. In this version, object and bin dimensions are depreciable. In other words, when bin capacity permits to assign more objects, they can be accommodated without any restriction regarding the height or length of the object.

The aim is to find a minimal L partition: $L = B_1 \cup B_2 \cup .... \cup B_m$. Such that, in each set B_j, the sum of the size of each object $w(a_i)$ in B_j does not exceed c. Where $1 \leq j \leq m$ [10].

4.2 Approximation Solution Algorithms

In this work, seven solution algorithms for BPP were used. Five of the algorithms are deterministic for specific purpose: First Fit Decreasing (FFD), Best Fit Decreasing (BFD), Match to First Fit (MFF), Match to Best Fit (MBF), and Modified Best Fit Decreasing (MBFD) [11, 12]. The remaining algorithms are non deterministic and of general purpose: Threshold Accepting (TA) [13] and Ant Colony Optimization (ACO) [14].

Information on algorithm performance for the nondeterministic algorithms was obtained calculating average performance for each instance in 30 executions, which is the minimum acceptable sample size [15].

5 Proposed Metrics for BPP Characterization

5.1 Problem Characterization

With the goal of modeling the structure of a instance, Pérez and Cruz [5] formulated the **SPI** (Specific-purpose Metrics) set, formed by five difficulty metrics: instance size (p), occupied capacity (t), dispersion (d), factors (f), and container usage (b).

In this work, two set of metrics based on *descriptive statistics* are proposed [16]. Different to the former metrics, these can be applied to a wide variety of problems.
GPI set (General- purpose Metrics); they characterize the weight distribution of BPP objects. Table 1 shows the 21 classical descriptive measures taken directly from statistics to define this set.

Table 1. Types of descriptive measures used to define the GPI set

Type	Measures
Centralization	Mean (arithmetic, harmonic and geometric), Median and Mode.
Dispersion	Range, Deviation (Sample Standard and Distribution Standard), Variance (Sample and Distribution), Variation Coefficient and Standard Error.
Position (Quantiles)	Deviation of ranges (Quartiles, Deciles and Percentiles).
Form	Skewness (Pearson's Coefficient - typical, median, mode -, Momental Skewness, Bowley Skewness) and Kurtosis.

GPI*c* **(**General-purpose Metrics scaled by the bin *c*apacity**).** This set is obtained from the GPI set. The weight of the objects w is scaled by the bin capacity c, as is showed in Equation 1.

$$w_i^c = \frac{w_i}{c}, \quad i = 1, 2, ..., n. \tag{1}$$

5.2 Algorithm Characterization

Two metrics were applied to determine solution quality: the absolute and the theoretical ratios. The first was used on instances that have solutions reported in the literature as the best found. The second was applied to instances that do not have a reference solution.

Equation 2 corresponds to the absolute ratio r_a, which is the ratio between, the number of bins of the obtained solution z_{obt}, and the number of bins of the best reference solution z_{obt}.

$$r_a = \frac{z_{obt}}{z_{best}} \tag{2}$$

Theoretical ratio r_t, formulated in Equation 3, is one of the most common metrics for Bin Packing: the ratio between z_{obt} and the theoretical number of bins z_t. This latter measure is a lower bound of the optimal bin number, and is equal to the sum of the object sizes divided by bin capacity (see Equation 4).

$$r_t = \frac{z_{best}}{z_t} \tag{3}$$

$$z_t = \frac{\sum_{i=1}^{n} w_i}{c} \tag{4}$$

In order to determine the best-performing algorithm for a given instance, greater priority was given to ratios than to run time.

5.3 Transformation from Multiclass Instances to Single-Class Instances

The proposed methods for transforming multiclass instances to single-class instances are described below.

EL (*Encapsulation of Labels*): This method considers each different class appearing in the dataset as a new class. A similar method called n-Model has been used by Boutell [16] and Wieczorkowska [18].

ELP (*Encapsulation of Labels with Partitioning*). First, initial groups of instances are defined using the classes generated with the EL method. Each group is partitioned using the K-Means method [19]. If the distance between the centers of the subgroups overcomes a threshold, set in this work at 0.5, the partition is conserved. The obtained groups are labeled as new classes, derived from the original classes.

VE-SI (**V**ertical **E**limination - **S**ingle-Label **I**nstances). A common transformation method is to eliminate all multiclass instances (horizontal elimination). VE-SI is an alternative proposal to eliminate classes (vertical elimination). For each instance, only the first associated class Y is conserved, where Y is a set of classes that occurs for single-class instances. For those instances with several classes in Y, one of their classes is select at random.

As can be observed, transformation methods hide information for the construction of the prediction models. Methods EL and ELP encapsulate class combinations, taking them as new classes. The VE-SI keeps only one class for each class combination.

6 Experimentation

The goal of the experiments is to analyze how the proposed measures impact the prediction accuracy. In particular, for algorithm selection applied to the BPP, in the context of a Hybrid Solution System. For the experimentation, two sets of optimization instances were used. Through the characterization process described in Section 5, the optimization instances were transformed into learning instances. The learning instances of BPP used to build the algorithm selector are called ASBPP (Algorithm Selection for Bin-Packing Problem). The ASBPP-training set is formed by 2,430 random BPP instances and performance information [5] of seven solution algorithms. The ASBPP-test set is formed by 1,369 standard BPP instances [20, 21] and information from the seven algorithms previously mentioned.

To build a prediction model of algorithm performance, the classification technique C4.5 of Quinlan [22] was applied to ASBPP-training. For model validation, two methods were applied, that are described widely in Dallas [23]: Resubstitution Validation (RV) and Validation with the Test Dataset (VTD).

To determine the prediction-model accuracy for a given instance, Pérez and Cruz [5] established the following criterion: the instance is well-classified if there is a common class between the set of predicted classes and the set of exact classes.

7 Results

Table 2 shows the accuracy results for the algorithm selector when applying the three transformation methods (EL, ELP and VE-SI) described in Section 5.3. We observed

that with the VTD validation method the SPI-GPI*c* set, applying the ELP transformation method obtains better performance; the same behavior is observed for the RV validation method. The incorporation of general-purpose metrics permits to improve on the 76% accuracy reported for the VTD in previous work [5].

Table 2. Results of the methods of validation with the ELP transformation

Transformation Method	Indice Set	Accuracy (%)	
		RV	VTD
EL	SPI	88.8066	77.0636
	SPI-GPI*c*	92.0165	76.6983
ELP	SPI	88.0247	76.0636
	SPI-GPI*c*	93.3745	**79.4010**
VE-SI	SPI	84.8560	68.8824
	SPI-GPI*c*	87.8189	*70.2703*

The results obtained with the EL and the ELP transformation methods are an upper bound for the exact result. In order to make a fairer evaluation, it was necessary to apply the VE-SI transformation method. For both validation methods RV and VTD, the SPI-GPI***c*** obtains higher accuracy percentage than other set. The transformation method VE-SI corresponds to realistic conditions. Therefore, in experimental conditions oriented to practical aims, the SPI-GPI*m*, which is composed by specific and general purpose metrics scaled by mean, obtained a greater accuracy with both validation methods.

8 Conclusions

In this work, a characterization process to extract learning instances from a set of optimization instances was proposed. In order to demonstrate the feasibility of the proposal, the Bin Packing Problem, solved with seven approximation solution algorithms, was characterized. Learning instances of Bin Packing were used in the construction of performance-prediction models. The models were validated with a set of standard instances aiming to quantify the impact of incorporating new characterization metrics to existent metrics developed in previous work [5].

For characterization of Bin Packing Problem, metrics derived from classical statistical measures are proposed. These metrics are of general purpose, because they are independent of the problem and can be applied to other optimization problems. Each metric is easily computed and represented by a numerical value.

For model construction, the C4.5 learning technique was applied. Like other classifiers, it does not support instances with more than one class associated (dominant algorithm). For this reason it was necessary to develop three techniques to transform instances with more than one associated algorithm: EL, ELP, and VE-SI transformations. The first two encapsulate an algorithm set into one class, and the last eliminates algorithms, preserving only one. Also, a transformation is necessary for setups where, for a given instance, only one solution algorithm can be executed.

Previous work reported a prediction accuracy of 76%, using the C4.5 and the Test Dataset validation method. Under the same conditions, an increase of 3.4% was obtained by incorporating a set of general-purpose metrics GPI*c* and applying the transformation technique ELP. The EL and ELP produce a model prediction that infers one or more algorithms for a given instance. The VE-SI produces a prediction model that infers a single algorithm for a given instance. For this reason, the VE-SI provides an lower bound of the exact result for prediction model accuracy when the EL and ELP transformations are applied. An accuracy of 70.27% was obtained by using VE-SI and SPI-GPI*c*, which represents an adequate result for real-word life conditions.

References

1. Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Mineola. Dover Publications, New York (1998)
2. Reeves, C.R.: Modern heuristic techniques for combinatorial problems. John Wiley & Sons, New York (1993)
3. Smith-Miles, K., James, R., Giffin, J., Tu, Y.: Understanding the relationship between scheduling problem structure and heuristic performance using knowledge discovery. In: Learning and Intelligent Optimization, LION, vol. 3 (2009)
4. Messelis, T., Haspeslagh, S., Bilgin, B., De Causmaecker, P., Vanden Berghe, G.: Towards prediction of algorithm performance in real world optimisation problems. In: Proceedings of the 21st Benelux Conference on Artificial Intelligence, BNAIC, Eindhoven pp. 177–183 (2009)
5. Pérez, J., Pazos, R.A., Frausto, J., Rodríguez, G., Romero, D., Cruz, L.: A Statistical Approach for Algorithm Selection. In: Ribeiro, C.C., Martins, S.L. (eds.) WEA 2004. LNCS, vol. 3059, pp. 417–431. Springer, Heidelberg (2004)
6. Borghetti, B.J.: Inference algorithm Performance and Selection under Constrained Resources. M.Sc. Thesis. Faculty of the School of Engineering Air University Force Institute of Technology Air University (1996)
7. Yuan, B.: Towards Improved Experimental Evaluation and Comparison of Evolutionary Algorithms. Ph.D. Thesis, School of Information Technology and Electrical Engineering of The University of Queensland, Australia (2006)
8. Gagliolo, M., Schmidhuber, J.: Learning Dynamic Algorithm Portfolios. In: AI & MATH 2006, Special Issue of the Annals of Mathematics and Artificial Intelligence, vol. 47, pp. 295–328. Springer, Heidelberg (2007)
9. Leyton-Brown, K., Nudelman, E., Shoham, Y.: Learning the Empirical Hardness of Optimization Problems: the case of combinatorial auctions. In: Van Hentenryck, P. (ed.) CP 2002. LNCS, vol. 2470, pp. 556–572. Springer, Heidelberg (2002)
10. Coffman, E.G., Courboubetis, C., Garey, M.R., Johnson, D.S., Shor, P.W., Weber, R.R.: Perfect Packing Theorems and the Average Case Behavior of Optimal and Online Bin Packing. ACM-SIAM Review. Society for Industrial and Applied Mathematics 44, 95–108 (2002)
11. Coffman, J., Garey, M., Jonson, D.: Approximation Algorithms for Bin-Packing, a Survey. In: Approximation Algorithms for NP-hard Problems, pp. 46–93. PWS, Boston (1997)
12. Coffman, J.E., Galambos, G., Martello, S., Vigo, D.: Bin Packing Approximation Algorithms: Combinatorial Analysis. In: Du, D.-Z., Pardalos, P.M. (eds.) Handbook of Combinatorial Optimization, pp. 151–207. Kluwer Academic Publishers, Dordrecht (1999)

13. Pérez, J., Pazos, R.A., Vélez, L., Rodríguez, G.: Automatic Generation of Control Parameters for the Threshold Accepting Algorithm. In: Coello, C.A., Albornoz, A., Sucar, L.E., Cairó, O.B. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 119–127. Springer, Heidelberg (2002)
14. Ducatelle, F., Levine, J.: Ant Colony Optimisation for Bin Packing and Cutting Stock Problems. In: Proceedings of the UK Workshop on Computational Intelligence, Edinburgh (2001)
15. Ross, S.M.: Simulación. Segunda edición. Prentice Hall, EEUU (1999)
16. Weisstein, E.W.: CRC Concise Encyclopedia of Mathematics, 2nd edn. CRC Press, FL (2002)
17. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning Multi-label Scene Classification. Pattern Recognition 37, 1757–1771 (2004)
18. Wieczorkowska, A., Synak, P., Ras, Z.: Multi-label classification of emotions in music. In: Proceedings of IIS 2006 Symposium, Intelligent Information Processing and Web Mining, Advances in Soft Computing, pp. 307–315. Springer, Heidelberg (2006)
19. Alsabti, K., Ranka, S., Singh, V.: An Efficient K-Means Clustering Algorithm. In: IPPS/SPDP Workshop on High Performance Data Mining, Orlando, Florida (1998)
20. Beasley, J.E.: Bin Packing - one dimensional. Brunel University. The Beasley's OR-Library, http://people.brunel.ac.uk/~mastjjb/jeb/orlib/binpackinfo.html
21. Scholl, A., Klein, R.: Bin Packing, Jena Univerity, http://www.wiwi.uni-jena.de/Entscheidung/binpp/metric.htm
22. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo; Book Review, vol. 16(3), pp. 235–240. Springer, Heidelberg (1993)
23. Dallas, J.: Métodos Multivariados Aplicados al Análisis de Datos. In: Thomson (ed.) Internacional, México (2000)

Improving Classification Performance of BCIs by Using Stationary Common Spatial Patterns and Unsupervised Bias Adaptation

Wojciech Wojcikiewicz[1,2,3], Carmen Vidaurre[1], and Motoaki Kawanabe[2,1,⋆]

[1] Technical University of Berlin, Franklinstr. 28 / 29, 10587 Berlin, Germany
wojwoj@mail.tu-berlin
[2] Fraunhofer Institute FIRST, Kekuléstr. 7, 12489 Berlin, Germany
carmen.vidaurre@tu-berlin.de
[3] Bernstein Center for Computational Neuroscience, Philippstr. 13, 10115 Berlin, Germany
nabe@first.fraunhofer.de

Abstract. Non-stationarities in EEG signals coming from electrode artefacts, muscular activity or changes of task involvement can negatively affect the classification accuracy of Brain-Computer Interface (BCI) systems. In this paper we investigate three methods to alleviate this: (1) Regularization of Common Spatial Patterns (CSP) towards stationary subspaces in order to reduce the influence of artefacts. (2) Unsupervised adaptation of the classifier bias with the goal to account for systematic shifts of the features occurring for example in the transition from calibration to feedback session or with increasing fatigue of the subject. (3) Decomposition of the CSP projection matrix into a whitening and a rotation part and adaptation of the whitening matrix in order to reduce the influence of non-task related changes. We study all three approaches on a data set of 80 subjects and show that stationary features with bias adaptation significantly outperforms the other combinations.

Keywords: Brain-Computer Interface, Common Spatial Patterns, stationary features, adaptive classification.

1 Introduction

Brain-Computer Interface (BCI) systems [4] aim to translate the intent of a subject measured from brain signals e.g. EEG into control commands for a computer application or a neuroprosthesis. A popular paradigm for BCI communication is motor imagery i.e. subjects perform the imagination of movements with their feet

⋆ We thank Klaus-Robert Müller for valuable discussions. This work was supported by the German Research Foundation (GRK 1589/1), the European Union under the project TOBI (FP7-ICT-224631) and the Federal Ministry of Economics and Technology of Germany under the project THESEUS (01MQ07018). This publication only reflects the authors' views. Funding agencies are not liable for any use that may be made of the information contained herein.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 34–41, 2011.

or hands, the imagined movements are detected by the system and translated into computer commands. The detection step often involves the extraction of relevant features by a spatial filtering method called Common Spatial Patterns (CSP) [1]. The CSP filter is computed during the calibration session, but can be adversely affected by noise and non-stationarities e.g. coming from changes in impedance, muscular activity or eye movements. Moreover, even optimal filters may result in bad classification accuracies as the distribution of the features can change over time due to differences between sessions e.g. no feedback vs. feedback or changes of task involvement. Therefore this paper aims at both, we propose a regularized version of CSP to alleviate the impact of bad trials and noisy electrodes and we combine it with methods which adapt to changes in the feedback session.

Recently, several approaches were proposed to reduce the impact of non-stationarities in BCI applications. For example [10] uses techniques for co-adaptive learning of user and machine, [9] investigates different methods for unsupervised adaptation of the classifier and [8] uses adaptive spatial filtering. Other approaches use extra measurement like EOG or EMG to remove artefacts [2] or apply covariate shift adaptation to account for shifts of the features [5].

This paper is organized as follows. In Section 2 we present stationary CSP and the adaptation methods. Section 3 describes the experimental setup. After that we present and analyse the results in Section 4. Finally we concludes with a short summary and future research ideas.

2 Methods

Stationary CSP

Stationary Common Spatial Patterns (sCSP) is inspired by invariantCSP [2,6]. However, in contrast to the latter it is completely data driven without using neurophysiological prior knowledge or extra measurements. The objective function of sCSP trades-off discriminativity and stationarity as it maximizes the variance of one class while minimizing the variance of the other class, but at the same time it penalizes non-stationary directions:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^\top \boldsymbol{\Sigma}_+ \mathbf{w}}{\mathbf{w}^\top \{\boldsymbol{\Sigma}_+ + \boldsymbol{\Sigma}_- + \lambda \overline{\boldsymbol{\Delta}}\} \mathbf{w}}, \tag{1}$$

$$\max_{\mathbf{w}} \frac{\mathbf{w}^\top \boldsymbol{\Sigma}_- \mathbf{w}}{\mathbf{w}^\top \{\boldsymbol{\Sigma}_+ + \boldsymbol{\Sigma}_- + \lambda \overline{\boldsymbol{\Delta}}\} \mathbf{w}}. \tag{2}$$

Note that $\mathbf{w}$ is the sCSP filter, $\boldsymbol{\Sigma}_+$ and $\boldsymbol{\Sigma}_-$ are average covariance matrices of the two classes, $\overline{\boldsymbol{\Delta}}$ is the regularization term and λ is a trade-off parameter.

In order to estimate the regularization term $\overline{\boldsymbol{\Delta}}$, we first compute the trial-wise covariance matrices $\boldsymbol{\Sigma}_+^{(k)}$ and $\boldsymbol{\Sigma}_-^{(k)}$ for both classes. After that we compute the deviations of the trial-wise covariance matrices from the class-average covariance matrix i.e. $\boldsymbol{\Delta}_i^{(k)} := \mathcal{P}\left(\boldsymbol{\Sigma}_i^{(k)} - \boldsymbol{\Sigma}_i\right)$ with i being the class indicator. Note that $\mathcal{P}$ is an operator to force symmetric matrices be positive definite. This assures

that the penalty term is always positive. In our study we simply flip the sign of negative eigenvalues. The regularization term $\overline{\mathbf{\Delta}}$ is finally defined as the average of the difference matrices $\mathbf{\Delta}_i^{(k)}$ over all trials k and the two classes i.

Unsupervised Bias Adaptation

In this paper we use the bias adaptation method PMean introduced in [9]. This method adapts the bias b of an Linear Discriminant Analysis (LDA) classifier $\mathbf{w}^T\mathbf{x} + b(t)$ by updating the global mean $\boldsymbol{\mu}$ of the features:

$$b(t) = \mathbf{w}^T \boldsymbol{\mu}(t-1), \tag{3}$$

$$\boldsymbol{\mu}(t) = (1-\eta) \cdot \boldsymbol{\mu}(t-1) + \eta \cdot \mathbf{x}(t), \tag{4}$$

where t is the trial number, $\eta \in [0\ 1]$ is a constant, $\mathbf{w}$ are the weights of the LDA classifier and $\mathbf{x}$ is the feature vector. Note that the adaptation is only performed in the feedback session and only affects the bias and not the weights.

Whitening Adaptation

The main idea of whitening adaptation is to decompose the CSP projection into a whitening $\mathbf{P}$ and a rotation $\mathbf{B}$ part. The rotation part is assumed to be fixed, but the whitening matrix is updated during feedback. This idea was introduced in [8]. There the covariance matrix was updated using blocks of data. In this work, we update the covariance using the same formula as for PMean, i.e. equation (4), because this approach only needs to store one sample rather than a block of data and allows a continuous adaptation:

$$\mathbf{P}(t) = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T, \tag{5}$$

$$\mathbf{\Sigma}(t) = (1-\eta) \cdot \mathbf{\Sigma}(t-1) + \eta \cdot \mathbf{\Sigma}^t, \tag{6}$$

where t is the trial number, $\mathbf{V}$ and $\mathbf{D}$ are the eigenvectors and eigenvalues of $\mathbf{\Sigma}(t-1)$, $\eta \in [0\ 1]$ is a constant and $\mathbf{\Sigma}^t$ is the covariance matrix of trial t. As before the adaptation is only performed in the feedback session and is unsupervised.

3 Experimental Setup

The experiments are based on data from a joint study [3] with Univesity Tübingen. We asked 80 volunteers to perform motor imagery tasks with the left and right hand or with the feet. For each user we select the best binary task-combination and estimate parameters like frequency band or time interval of interest in a calibration session (150 trials) without providing feedback. After that we perform a test session consisting of 300 trials and provide 1D visual feedback i.e. a moving arrow on a screen. All subjects in this study are BCI novices. We use recordings of 68 preselected electrodes densely covering the motor cortex, three CSP directions per class, power features, an LDA classifier and

error rate to measure performance. We select the λ parameter for sCSP from the set of candidates $\{0, 0.1, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1, 2.5, 5, 10\}$ by 5-fold cross-validation on the calibration data. As the authors in [9] we use $\eta = 0.05$ for PMean. For the whitening adaptation η is chosen to be one order of magnitude smaller i.e. 0.005 as the method is more sensitive to changes.

4 Results

At first we compare the performances of CSP and sCSP with and without using adaptation. Fig. 1 shows six scatter plots, each displaying the error rates of two methods which we want to compare. We see that in general sCSP performs much better than CSP, although for some subjects it can lead to slightly worse results. We will analyse the reasons for the performance increase further below. We also see that using bias adaptation improves classification results as it reduces shifts in the features which can occur between calibration and test session[1]. However, except for a few users PMean is superior to whitening adaptation (see right bottom panel). We also investigated the combination of sCSP with both adaptation methods, but it did not bring any significant improvements over sCSP and PMean. We also observed unequal false positive and false negative rates when using CSP or sCSP. This bias comes from a shift in the distribution of features between calibration and test phase and can be resolved by bias adaptation.

In order to evaluate the significance of our results we use the Wilcoxon signed-rank test. Table 1 shows the p-values (one-sided) of different comparisons using either all subjects or dividing them into three groups based on their error rates: 0% - 15%, 15% - 30% and above 30%. Note that the null hypothesis of the test states that the median of the distribution of error rate differences is zero. Together with the scatter plots from Fig. 1 we can not only tell whether there is a significant difference between two methods or not, but also which one performs better. Most significant improvements are obtained for subjects in the above 30% error rate group. For these participants both using stationary features and adapting to changes is advantageous, whereas users who perform well do not benefit from sCSP. This is intuitive as subjects with low error rates usually have a clean signal i.e. it is not affected by artefacts and the signal to noise ratio is higher than in subjects who lack BCI efficiency. Therefore using sCSP does not bring any advantage. However, since these subjects can also suffer from shifts in the features (as introduced by changing from calibration to feedback condition), adaptation methods may improve the classification results.

In the following we would like to analyse the reasons for the performance gain on a specific participant. This subject performs left vs. right motor imagery and has an error rate of 28.67% when using CSP. However, the error rate decreases to 8% when applying sCSP (with $\lambda = 0.25$), it is 6% for both sCSPpmean and sCSPwhitening and it is 10.33% for CSPpmean and 9.67% for CSPwhitening. So why does CSP perform so poorly compared to the other methods?

[1] Note that visual feedback was only provided in the test session.

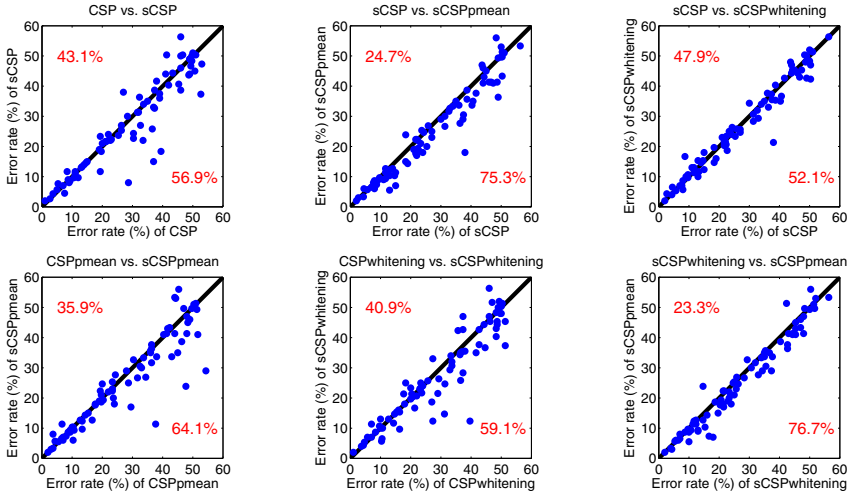


Fig. 1. Six scatter plots comparing the error rates of different methods. Each subject is represented by a blue dot. The percentage of points lying above or below the black line is shown in red and indicates which of the compared methods performs better. We see that both sCSP and bias adaptation, especially PMean, decrease error rates.

Table 1. Overview of p-values for different method combinations using Wilcoxon signed-rank test (one-sided). Bold values are significant when $\alpha = 0.05$. Grouping is performed based on the error rates of the worse method (x-axis in Fig. 1).

Compared Methods	Groups based on error rate (in %)			
	$0-15$	$15-30$	>30	all
sCSP better than CSP	0.4924	0.4510	**0.0118**	**0.0301**
sCSPpmean better than sCSP	**0.0033**	**0.0076**	**0.0006**	**0**
sCSPwhitening better than sCSP	0.4925	**0.0098**	**0.0103**	**0.0022**
sCSPpmean better than CSPpmean	0.2429	0.2035	**0.0192**	**0.0117**
sCSPwhitening better than CSPwhitening	0.2146	0.4691	**0.0089**	**0.0138**
sCSPpmean better than sCSPwhitening	0.0605	**0.0011**	**0.0130**	**0**
sCSPpmean better than CSP	**0.0026**	**0.0014**	**0.0022**	**0**

In order to answer this question we will first identify the dimensions which are responsible for the bad performance of CSP. It seems that two out of the six CSP filters are corrupted by noise and non-stationarities and degrade the overall performance since when classifying the data without dimensions 2 and 3, the error rate of CSP decreases to 10%. If on the other hand we include one of the noisy dimensions, the error rate increases to more than 20%. Fig. 2 shows the filters of dimension 2 and 3 for CSP and sCSP using the same scale. We see that the CSP filters heavily weight electrodes at different locations including areas which are primarily not responsible for performing motor imagery like frontal

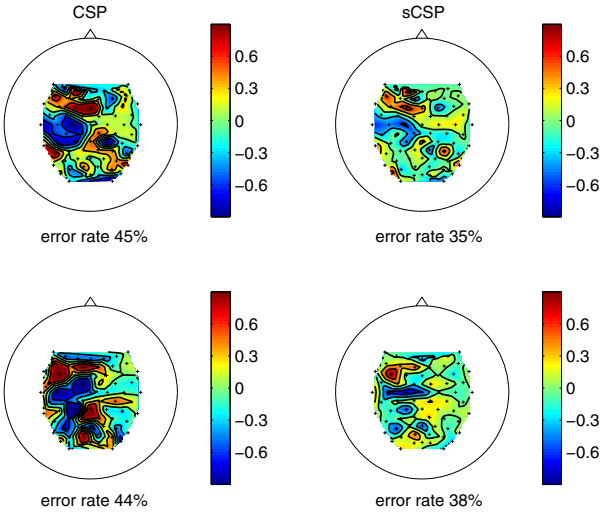


Fig. 2. Filters of noisy dimensions 2 (top) and 3 (bottom) for CSP and sCSP

and temporal areas. Since the filters are obviously corrupted, the error rates of the corresponding features are almost at chance level. In contrast to that the corresponding sCSP filters are much less affected by artefacts in the data which leads to a more reasonable weighting of the channels and smaller error rates.

A remaining question is what effect does the reduction of the weights have on the features. In Fig. 3 we plot the calibration and test features of the noisy dimensions 2 and the most discriminant[2] dimension 4 for CSP and sCSP. As can be seen from the left panel there is a significant shift in dimension 2 between calibration and test features. Since the influence of bad trials and noisy electrodes is reduced when using sCSP, this shift is much weaker in the right panel. So we conclude that for this subject non-stationarities in the data, most probably eye movements and some noisy electrodes, have corrupted two CSP filters and thus lead to shifts in the features. These shifts can be either removed by applying adaptation methods like PMean or by computing stationary features with sCSP.

If we analyse Fig. 4 we see that using whitening adaptation also reduces the shift in dimension 2 and 3. This figure shows the mean absolute difference between calibration and test features for each dimension after a different number of updates of the covariance matrix. The reduction of the shift in dimension 2 and 3 is probably the main reason for the performance gain of CSPwhitening. However, whitening adaptation does not only reduce the shift between dimensions but can also increase it e.g. for dimension 6. We conjecture that this may be a reason why whitening adaptation is inferior to PMean.

[2] The error rate of this dimension is 16% for CSP and 17% for sCSP.

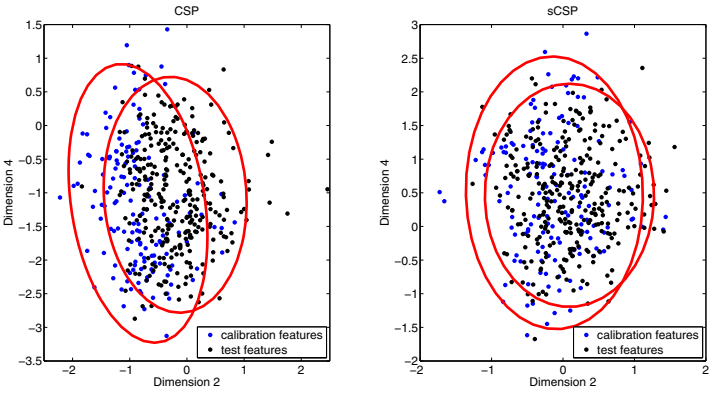


Fig. 3. Visualization of feature dimension 2 and 4 for CSP and sCSP. The shift between calibration and test features in dimension 2 is due to eye movements artefacts which adversely affect the CSP filter, but have less impact when using sCSP.

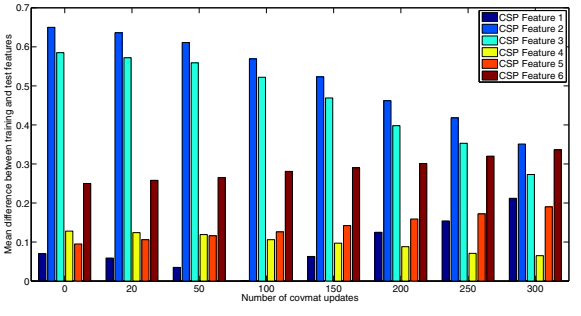


Fig. 4. Mean absolute differences between calibration and test features for different feature dimensions after a different number of update steps of the covariance matrix.

We would further like to note that not in all cases sCSP and PMean have the same effect, namely the reduction of the shift in the features. For some subjects applying sCSP helps a lot, whereas CSPpmean or CSPwhitening does not perform any better than CSP. In these subjects the influence of noise and non-stationarities is often so strong that all CSP filters are more or less corrupted. Thus adaptation methods can not improve results as no dimension is discriminative. In our example on the other hand there is a discriminative dimension after application of CSP, but other noisy dimensions lead to a shift in the features and affect the performance negatively. In both cases sCSP reduces the influence of noisy and non-stationarity directions and can improve classification accuracy. However, since sCSP is computed on calibration data it is not able to capture changes which does not appear during this phase. Therefore combining sCSP with PMean provides best results as it reduces the adverse effects on non-stationarities in both the calibration and the test session.

5 Conclusion

We presented an extension of CSP which explicitly measures non-stationarities and regularizes the CSP directions towards stationary subspaces. We showed that the main reason why sCSP improves performance is that it reduces the influence of bad trials (artefacts) and noisy electrodes which can corrupt the CSP filters. Unlike other methods, such as invariantCSP, our method is completely data-driven and does not need additional recordings or models of the expected change that occurs in the EEG. However, it cannot capture changes which occur only in the feedback session since sCSP is computed on calibration data. Therefore further methods of adaptation are required. We showed that combining sCSP with PMean significantly outperforms the baseline and is also superior to sCSP with whitening adaptation.

In future research we would like to investigate other data-driven regularization criteria and compare sCSP to invariantCSP with additional EOG and EMG recordings.

References

1. Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., Müller, K.-R.: Optimizing spatial filters for robust eeg single-trial analysis. IEEE Signal Proc. Magazine, 581–607 (2008)
2. Blankertz, B., Kawanabe, M., Tomioka, R., Hohlefeld, F.U., Nikulin, V., Müller, K.-R.: Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing. Ad. in NIPS 20 (2008)
3. Blankertz, B., Sannelli, C., Halder, S., Hammer, E.M., Kübler, A., Müller, K.-R., Curio, G., Dickhaus, T.: Neurophysiological Predictor of SMR-Based BCI Performance. NeuroImage 51(4), 1303–1309 (2010)
4. Dornhege, G., Millán, J.d.R., Hinterberger, T., McFarland, D., Müller, K.-R.: Toward Brain-Computer Interfacing. MIT Press, Cambridge (2007)
5. Li, Y., Kambara, H., Koike, Y., Sugiyama, M.: Application of covariate shift adaptation techniques in brain-computer interfaces. IEEE Trans. Biomed. Eng. 57(6), 1318–1324 (2010)
6. Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A., Müller, K.-R.: Invariant feature extraction and classification in kernel spaces. Ad. in NIPS 12, 526–532 (2000)
7. Ramoser, H., Müller-Gerking, J., Pfurtscheller, G.: Optimal spatial filtering of single trial EEG during imagined hand movement. IEEE Trans. Rehab. Eng. 8, 441–446 (1998)
8. Tomioka, R., Hill, J.N., Blankertz, B., Aihara, K.: Adapting Spatial Filter Methods for Nonstationary BCIs. In: Proc. of Workshop on Information-Based Induction Sciences (IBIS), pp. 65–70 (2006)
9. Vidaurre, C., Kawanabe, M., von Bünau, P., Blankertz, B., Müller, K.-R.: Toward an unsupervised adaptation of LDA for Brain-Computer Interfaces. IEEE Trans. Biomed. Eng. (2010) (in press)
10. Vidaurre, C., Sannelli, C., Müller, K.-R., Blankertz, B.: Machine-learning based co-adaptive calibration. Neural Comput. (2010) (in press)

A Simple Proactive Provider Participation Technique in a Mesh-Based Peer-to-Peer Streaming Service

Darío Padula, María Elisa Bertinat, Franco Robledo Amoza, Pablo Rodríguez-Bocca, and Pablo Romero

Laboratorio de Probabilidad y Estadística, Facultad de Ingeniería
Universidad de la República. Montevideo, Uruguay
dpadula@fing.edu.uy
www.lpe.edu.uy

Abstract. The design of efficient protocols for mesh-based Peer-to-Peer (P2P) networks has many challenges, one of them is bandwidth allocation. On one hand, peers demand high Quality of Experience and network traffic when they watch their streaming contents. On the other, Internet Service Providers (ISPs) support their business with the capacity of their international links. A recent strategy considered in order to meet both requirements is the Proactive Provider Participation, shortly named P4P [1], [2]. This paper addresses the multi-objective P4P problem for live video content. First, we introduce a Ford-Fulkerson-based solution, that solves the P4P problem when only one content is distributed. Second, a greedy randomized technique is applied when multiple contents are shared. Finally, our algorithm is tested in a real multiple-content scenario, showing that the results highly outperform current strategies.

Keywords: Peer-to-Peer, P4P, Network Optimization.

1 Introduction

An important amount of today's Internet traffic is due to P2P networks, particularly for live video streaming [3,4]. For this reason, several peer-to-peer streaming networks were developed in the last years. The most successful ones are PPlive, TVUnetwork, SopCast, with proprietary and unpublished mesh-based protocols [3]. These are virtual networks developed at the application layer, over the Internet infrastructure. The nodes, called *peers*, offer their resources to others, basically because they share interests in common. The users can connect and disconnect freely. This makes P2P networks an attractive tool for them, but increases P2P's design challenges, because the bandwidth resource fluctuates uncontrolled. In mesh-based protocols, the cooperation is the key element in order to assure a certain quality of experience to end-users [3]. There are three main steps in all mesh-based P2P protocols for cooperation. First, when a peer enters the net it should discover other peers sharing the same content, which is called *swarm selection strategy*. Once a new peer knows other peers in his swarm, he must select the best ones to cooperate, what is called *peer selection strategy*. Finally, it should decide which pieces of the streaming content should be asked first, called the *piece selection strategy* [5], [6]. This paper is focused on the swarm selection strategy and in the

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 42–50, 2011.

peer selection strategy. The main issue is to allocate the largest amount of traffic in the network without bottlenecks, and keeping the quality of experience between peers. In Sect. 2 the mathematical P4P model is explained (based on [1]). The reader can find related work on P4P in [7], [8], [9]. Section 3 contains a polytime resolution for one content. A Greedy Randomized heuristic [10] is proposed in Sect. 4 for the general case. In Sect. 5, we introduce this algorithm in a real P2P platform, called GoalBit [11].

2 Mathematical Model

This model is inspired in [1], where the authors simplify the problem into a linear programming one in order to solve it.In order to represent the complexity and scale of a real scenario with millions of peers, the peers are grouped in nodes. Each node is a geographical subset of Internet (for example: an autonomous system or an ISP point-of-presence), and they are interconnected by real links. Inside each node could be several peers sharing contents. Consider a network $N = (V, C)$ with nodes set $V = \{v_1, \ldots, v_n\}$, and a non-negative matrix with null diagonal $C = (c_{i,j})$ such that for each pair of different nodes v_i and v_j there are two one-way links, with capacities $c_{i,j}$ and $c_{j,i}$. The upload and download bandwidths are u_i^k and $d_i^k, i = 1, \ldots, n$ respectively, where $k \in \{1, \ldots, K\}$ represents different contents (each node v_i has K possible contents to download). Each link (i, j) uses a certain percentage of its capacity due to other applications, which is denoted by $b_{i,j}$ (called the background traffic). Be $\mathcal{P} = \mathcal{P}_1^{k_1}, \ldots, \mathcal{P}_m^{k_m}$ a set of oriented paths in the network, where $\mathcal{P}_h^{k_h} = (x_h, \ldots, y_h)$ (x_h is the uploader and y_h is the downloader). Be $t_1, \ldots, t_m$ their respective traffic magnitudes. In words: x_h uploads a traffic magnitude t_h of content type k_h to y_h by the oriented path $\mathcal{P}_h^{k_h}$. We assume that the intermediate nodes does not consume bandwidth. The ISPs objective (1) is to reduce the bottleneck over the most expensive edges in Internet, subject to the satisfaction of peers in the network (first constraint). The other constraints express that the set of oriented paths must be feasible (not exceeding the link's capacities neither the bandwidths). We will assume a practical *intra-domain traffic* hypothesis: $u_i^k = 0$ or $d_i^k = 0$ for each node v_i. Basically, the network works in steady state, where all the traffic uploaded is downloaded [1], [2].

$$\min_{\mathcal{P}} \max_{(i,j):i\neq j} \rho(\mathcal{P}) = b_{i,j} + \frac{\sum_{h:(i,j)\in\mathcal{P}_h^{k_h}} t_h}{c_{i,j}}, \qquad s.t. \tag{1}$$

P4P Model

$$\begin{cases} \max_{\mathcal{P}} \sum_{h=1}^{m} t_h & (2) \\ \sum_{h:x_h=i,k_h=k} t_h \leq u_i^k, \forall i \in V, k \in \{1, \ldots, K\} & (3) \\ \sum_{h:y_h=j,k_h=k} t_h \leq d_j^k, \forall j \in V, k \in \{1, \ldots, K\} & (4) \\ b_{i,j}c_{i,j} + \sum_{h:(i,j)\in\mathcal{P}_h}^{k_h} t_h \leq c_{i,j}, \forall i \neq j \in V & (5) \end{cases}$$

In practice, once we have a set of oriented paths $\mathcal{P}$ and their respective magnitudes, it is possible to converge probabilistically to that traffic distribution in a real network. In order to reach this goal the following strategies can be followed: the swarm for a peer located at node v_i, that asks for content k, is populated with the following proportion of peers from node v_j: $w_{ji}^k = \frac{\sum_{h:(j,i)\in\mathcal{P}_h^k} t_h}{\sum_{x\in V}\sum_{h:(x,i)\in\mathcal{P}_h^k} t_h}$. Moreover, the peer statistically takes in consideration these weights also in his peer selection strategy in order to have a faster converge. See [12] for details.

3 A Polytime Resolution for One Content

Although the high complexity of the general P4P formulation, we show there is a Fully Polynomial Time Approximation Scheme (FPTAS) when one content is distributed in the network (i.e. $K = 1$ in the P4P Model) in Sect. 3. These algorithm is naturally extended with a greedy randomized technique discussed in Sect. 4.

Definition 1. *An algorithm is a Fully Polynomial Time Approximation Scheme (FPTAS) for the optimization problem P if for any given instance I and $\epsilon > 0$, it finds in polynomial time in the input size I and in $\frac{1}{\epsilon}$ a solution S, which complies that $|val(S) - val(I)| < \epsilon \times val(I)$, where $val(I)$ is the optimal value for instance I.*

Algorithm 1. $\mathcal{P} = MaxFlow((V, C), B, u, d, \epsilon)$

```
1: C^net = C × (1 − B)
2: N_in = (V, C^net)
3: N_aux = Extend(s, t, N_in, u, d)
4: (P_M, φ_max) = FordFulkerson(N_aux)
5: ρ_min = max_{i≠j} b_ij
6: ρ_max = 1
7: while |ρ_max − ρ_min| > ε do
8:     ρ = (ρ_min + ρ_max)/2
9:     C^net = C × (ρ × 1 − B)
10:    Update(N_aux)
11:    (P, φ) = FordFulkerson(N_aux)
12:    if φ = φ_max then
13:        ρ_max = ρ
14:    else
15:        ρ_min = ρ
16:    end if
17: end while
18: return P
```

Algorithm $MaxFlow$ constructs an auxiliary network connecting every node of the original network $N = (V, C)$ with two ideal nodes s and t, and finds the minimum link utilization of the original network preserving at the same time the maximum flow (using Ford-Fulkerson Algorithm iteratively). It receives a network $N = (V, C)$, two vectors u and d that represent the upload and download bandwidths for each node, the background

matrix B and a given tolerance $\epsilon > 0$ from Definition 1. Lines $1-2$ substract the background traffic keeping the remainder C_{net} and define a network $N_{in} = (V, C_{net})$. In Line 3 an auxiliary network N_{aux} is constructed by extending N_{in} with Function $Extend$. It connects two ideal nodes s (the *source*) and t (the *terminal*) to N_{in}, where s (resp. t) is connected with every node $v_i \in V$ with its upload (download) capacity u_i (resp. d_i). Immediately in Line 4, a max-flow ϕ_{max} is found in N_{aux} between s and t, calling the classical Ford-Fulkerson Algorithm, where $\mathcal{P}$ is the set of output paths and their flow magnitudes (recall that Ford Fulkerson returns the min-cut, but a maximal flow can be saved as well). Next, the main idea is to find the minimum percentage link utilization ρ that achieves ϕ_{max}. The block of Lines $5-17$ consists of a bipartition search in the closed interval $[\rho_{min}, \rho_{max}]$, where $\rho_{max} = 1$ (all resources are used) and $\rho_{min} = max_{i \neq j} b_{ij}$ (no resources are used), until the desired tolerance ϵ is met. In each iteration, the capacities C^{net} of the auxiliary network N_{aux} are modified, according with ρ. Note that the logic of the bipartition preserves the maximum flow ϕ_{max}.

Theorem 1. *Algorithm $MaxFlow$ is a FPTAS for the P4P Problem when $K = 1$.*

Proof. Consider the set of output paths (with magnitudes) $\mathcal{P}$ in the auxiliary network N_{aux} and remove the ideal nodes s and t. The proof consists of three steps: feasibility, optimality and computational effort.

1. We assert that $\mathcal{P}$ *is a feasible set of paths for the P4P model*: In effect, it is clear that ϕ_{max} is preserved during the entire Algorithm $MaxFlow$, and by construction it is the maximum flow in the original network, satisfying constraint 2. Constraints 3 and 4 are fulfilled since the links capacities between s, t and every node v_i in the original network are chosen with respective capacities u_i and d_i. Constraint 5 is satisfied: set $\rho : \rho_{min} \leq \rho \leq 1$. The total flow for link (i, j) does not exceed its capacity, since Line 9 assures that $c^{net}_{i,j} = c_{i,j} \times (\rho - b_{ij})$ and:

$$b_{i,j}c_{i,j} + \sum_{h:(i,j)\in\mathcal{P}_h}^{k_h} t_h \leq b_{i,j}c_{i,j} + c^{net}_{i,j} \leq c_{i,j}\rho \leq c_{i,j},$$

   where $b_{i,j}c_{i,j}$ is the total traffic inside link (i, j) given to other applications.
2. Observe that $\mathcal{P}$ *is an optimal set of paths*. The Ford-Fulkerson algorithm finds the maximum flow between the nodes s and t in the auxiliary network N_{in}, which has a direct interpretation in the original network N.
3. Finally, $MaxFlow$ *is a polytime algorithm*: since Ford-Fulkerson is polynomial in the number of nodes n, it suffices to show that the number of iterations i in the bi-partition search is polytime in n and $1/\epsilon$. Given $\epsilon > 0$, there exists an integer i such that $2^i > \frac{1}{\epsilon}$. Consequently, $MaxFlow$ runs in polytime with n and in $i < \lfloor log_2 1/\epsilon \rfloor < 1 + \frac{1}{\epsilon}$ steps, which is a linear expression in $\frac{1}{\epsilon}$ as required in Definition 1. These inequalities hold for any given $\epsilon > 0$, so the global error can be bounded as well as the relative error.

In conclusion, $MaxFlow$ is a FPTAS for the P4P problem when one content is delivered, as we wanted to prove.

4 A Metaheuristic Approach for Multiple Contents

To the best of our knowledge, there is not an exact resolution in polytime for multiple contents [12]. A possible metaheuristic approach is detailed next.

Algorithm 2. $\mathcal{P}_{RP} = RandPriority(U, D, C, B, p)$

1: $\mathcal{P}_{RP} = \varnothing$
2: **while** $(length(p) > 0$ AND $C \geq 0)$ **do**
3: k = $ChooseContent(p)$
4: $\mathcal{P}_{RP} = \mathcal{P}_{RP} \cup MaxFlow((V, C), B, U^k, D^k)$
5: $Update(U, D, C, B, p)$
6: **end while**
7: **return** $\mathcal{P}_{RP}$

Algorithm $RandPriority$ is very simple, and proposes a Greedy Randomized generalization for multiple contents [10]. It receives the bandwidth matrices U and D (that store the bandwidth of every node for each content), the capacity and background matrices C and B respectively and a probability vector p_k that measures the priority to the content type k. In Line 1, the set of paths $\mathcal{P}_{RP}$ is empty. In each iteration, a content is chosen randomly according with the priority vector p (Line 3). Immediately, in Line 4 Algorithm $MaxFlow$ is called, in order to find the best routing for that content. The obtained flows are added to $\mathcal{P}_{RP}$, and the bandwidth and capacities updated in Line 5 (the entry k in the priority vector p is deleted, so content k is not considered any more). The process is repeated until there is no more capacity or after all contents were delivered. In order to get a reference of the performance achieved by $RandPriority$, it is desired to find an upper bound for the maximum traffic. We can easily translate every multiple-content instance into a songle-content one, identifying all contents as the same, and considering new upload and download bandwidths $u_i^{'}$ and $d_i^{'}$ for each node v_i [12]:

$$u_i^{'} = \sum_k u_i^k; \quad d_i^{'} = \sum_k d_i^k$$

In this way, the total traffic in the new single-content network ϕ_B is certainly not lower than the original traffic ϕ for the corresponding multiple content instance.

Definition 2. *Given an instance for the general P4P Model, the performance coefficient for an algorithm that achieves link utilization ρ and total traffic ϕ is defined by:*

$$\alpha = \rho \frac{\phi_B}{\phi}, \tag{6}$$

where ϕ_B is the traffic in the single-content network (obtained identifying contents) found applying Algorithm $MaxFlow$.

Algorithm $BestPaths$ combines these ideas, calling $RandPriority$ iteratively, and returning the best output (considering the performance coefficient, which is measured in each iteration).

Algorithm 3. $\mathcal{P}_{BP} = BestPaths(U, D, B, C, p, Iter, \phi_B)$

1: **for** $i = 1$ to $Iter$ **do**
2: $(\mathcal{P}_i) = RandPriority(U, D, B, C, p)$
3: $(\rho_i, \phi_i) = Calculate(\mathcal{P}_i)$
4: $\alpha_i = \rho_i \frac{\phi_B}{\phi_i}$
5: **end for**
6: $\alpha_j = min_{1 \leq i \leq Iter} \alpha_i$
7: **return** $\mathcal{P}_{BP} = \mathcal{P}_j$

5 Empirical Results

5.1 Performance in a Real Platform for One Content

In this work we implemented Algorithm $MaxFlow$ to converge empirically to the optimum P4P routing in a real P2P video streaming platform, called GoalBit [11], when only one content is distributed. It is known that BitTorrent works in downloading, but does not comply real time requirements for streaming applications [13, 6]. GoalBit maintains the BitTorrent's philosophy, trying to extend its success to video streaming. The routing protocol we propose acts exactly in the moment of the peer list conformation, applying $MaxFlow$ algorithm to skew routing to converge to the theoretical P4P solution. Emulations were carried out with information provided by The Uruguayan National Telephony Operator ANTEL[1] from their GoalBit deployed live service. We contrast the peer selection strategy using Algorithm $MaxFlow$ versus the Classical GoalBit strategy (based on BitTorrent) [11]. For all emulations we evaluate the quality of experience of final users (buffering time), the total amount of exchanged traffic and the one which crosses international links. In particular, we show the results of two emulations for the cases of 60 and 100 simultaneous peers connected in average.

Tables 1 and 2 show, for both strategies, the total traffic (incoming and outgoing international links traffic[2]). These tables also show the percentage growth of incoming and outgoing traffic $P_{G_{in}}$ and $P_{G_{out}}$ when the P4P model is applied, in relation with a Classical routing. Specifically:

$$P_{G_{in}} = 100 \times (\frac{In_{P4P}}{In_{Classic}} - 1); \quad P_{G_{out}} = 100 \times (\frac{Out_{P4P}}{Out_{Classic}} - 1),$$

where In_{P4P}, $In_{Classic}$ and Out_{P4P} and $Out_{Classic}$ represent the total incoming and outgoing traffic for P4P and Classic models respectively. It is desirable to obtain negative percentage growth, interpreted as a reduction in the international links, and consequently, an improvement in relation with the Classical routing strategy.

It can be appreciated from Table 1 that the incoming reduction is 47.57%, while the outgoing reaches 74.96%, keeping the total traffic achieved by the Classical routing. Also, Table 2 shows important reductions for the 100 simultaneous peers case, and an increasing in total traffic is perceived. Figure 1 shows that the buffering time distributions are quite similar for both techniques. In both cases, the 85% of peers perceived

[1] This paper was supported by The National Telephony Operator ANTEL (www.antel.com.uy).
[2] As we can only measure the traffic through the outgoing and incoming links to and from Uruguay, this will be our reference node.

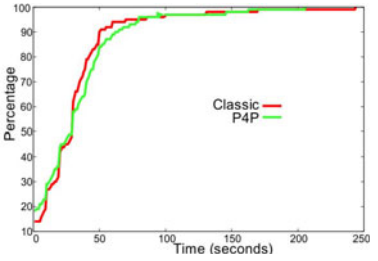


Fig. 1. Buffering time for 60 peers

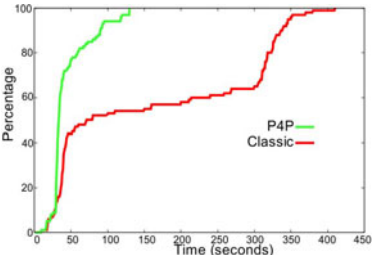


Fig. 2. Buffering time for 100 peers

Table 1. Link utilization for 60 peers

Model	*Incoming*	*Outgoing*	*Total*
Classic	31656	31366	183069
P4P	7927	16446	183067
$\%growtraffic$	-47.57	-74.96	0.0

Table 2. Link utilization for 100 peers

Model	*Incoming*	*Outgoing*	*Total*
Classic	5681	10253	55078
P4P	3657	4451	58893
$\%growtraffic$	-56.59	-35.63	6.93

a buffering time lower than 55 seconds. In contrast, for the 100 simultaneous peers case, P4P present much lower buffering times when compared with the Classical strategy (see Fig. 2). A 68% of peers wait no more than 38 seconds to start playing when P4P is applied, but only a 27% of peers can start playing the video during the same time. Many emulations were carried out for different inputs showing similar bandwidth savings, close to 30% in average [12].

5.2 Simulation for Multiple Contents

In order to understand the performance of Algorithm $BestPaths$, we developed a simple simulation, taking uniformly random bandwidth matrices $U = n \times Rand(n,k)$, $D = n \times r \times Rand(n,k)$, background and capacity matrices $B = 0$ and $C = k \times s \times Rand(n,n)$. The integers n and k are the number of nodes and contents in the network respectively; the factors $r > 1$ and $s > 1$ regulate how pessimistic or optimistic the scenario is. Note that an increment in these factors mean higher expected download bandwidths and capacities. We called $BestPaths$ taking $Iter = 10$ in order to find better results.

In Table 3 and Table 4 we specified (by the parameters n, k, r and s) a set of scenarios. Also, these tables show the results of the simulation. We ran 5 times each scenario in order to avoid random effects, and count the number of times the bound is reached (column $\%Success$ of Tables 3 and 4). They also show for each scenario the mean and standard deviation for: output link utilization ρ, (μ_ρ, σ_ρ), output traffic ϕ (μ_ϕ, σ_ϕ), and quotient between ϕ and the respective upper bound ϕ_B (called μ_{ϕ/ϕ_B} and σ_{ϕ/ϕ_B} respectively). It can be appreciated that the bound is reached in every optimistic instance (when $r = s = 2$). Interestingly, a success can be afforded in pessimistic scenarios also (see for example Row 3 of Table 4, when the resources were limited with $r = 1.2$ and $s = 1.5$, and the optimum was achieved in all instances).

Table 3. $BestPaths$ vs $Bound$ for different instances when $r = 2$ and $s = 2$

$Scenario$	n	k	r	s	μ_ρ	σ_ρ	μ_ϕ	σ_ϕ	$\mu_{\frac{\phi}{\phi_B}}$	$\sigma_{\frac{\phi}{\phi_B}}$	$\%Success$
1	20	5	2	2	0.9169	0.0959	1012.8	67.56	1.0000	0.0000	100
2	20	10	2	2	0.8022	0.0411	2040.4	54.16	1.0000	0.0000	100
3	40	5	2	2	0.8716	0.0807	4042.4	106.95	1.0000	0.0000	100
4	40	10	2	2	0.7361	0.0854	8068.3	278.12	1.0000	0.0000	100

Table 4. $BestPaths$ vs $Bound$ for different instances when $r = 1.2$ and $s = 1.5$.

$Scenario$	n	k	r	s	μ_ρ	σ_ρ	μ_ϕ	σ_ϕ	$\mu_{\frac{\phi}{\phi_B}}$	$\sigma_{\frac{\phi}{\phi_B}}$	$\%Success$
1	20	5	1.2	1.5	0.9907	0.0207	1008.7	31.83	0.9961	0.0043	20
2	20	10	1.2	1.5	0.9651	0.0482	1943.0	105.90	0.9978	0.0028	40
3	40	5	1.2	1.5	1.0000	0.0000	4036.8	125.42	1.0000	0.0000	100
4	40	10	1.2	1.5	0.9500	0.0314	7935.9	212.31	0.9952	0.0057	20

6 Conclusions

In this work, the Proactive Provider Participation (P4P) model was analyzed. When all peers are interested in exactly one content, the problem can be solved with the desired accuracy, combining a Ford-Fulkerson approach and a bi-partition scheme. However, a greedy randomized technique was developed for the general case, inspired in the success for single content distribution. It selects contents randomly and allocates bandwidth according with the previously methodology for one content. Emulations contrasting a bandwidth allocation in a real platform (GoalBit) and the P4P resolution are very challenging. They indicate a 30% link utilization reduction in average with our P4P application, and at the same time the quality of experience seems to improve. We carried out simulations for optimistic and pessimistic scenarios when multiple contents are delivered. An upper bound for the maximum traffic was introduced, identifying all contents as the same. Every optimistic instance achieves 100% success, meaning that the maximum flow is sent at a minimum link utilization running the greedy randomized algorithm for multiple contents. This ideal upper bound is also achieved in every pessimistic scenario, at least once. Qualitatively, this highlights the competitiveness of the P4P model for bandwidth allocation.

References

1. Xie, H., Krishnamurthy, A., Silberschatz, A., Yang, Y.R.: P4P: Proactive Provider Participation for P2P. Technical report, Yale University, Department of Computer Science (2007)
2. Xie, H., Yang, R., Krishnamurthy, A., Liu, Y., Silberschatz, A.: P4P: Provider portal for applications. In: Conference on Data Communication, SIGCOMM (2008)
3. Rodríguez-Bocca, P.: Quality-centric design of Peer-to-Peer systems for live-video broadcasting. PhD thesis, INRIA/IRISA, Université de Rennes I, France (April 2008)
4. Cisco Systems, Inc. Hyperconnectivity and the Approaching Zettabyte Era (June 2010)

5. Bertinat, M.E., Vera, D.D., Padula, D., Robledo, F., Rodríguez-Bocca, P., Romero, P., Rubino, G.: A COP for Cooperation in a P2P Streaming Protocol. In: Proceedings of the IEEE International Conference on Ultra Modern Telecommunications (ICUMT 2009), October 12-14, pp. 1–7. IEEE Computer Society, Washington (2009)
6. Romero, P.: Optimización de la Estrategia de Selección de Piezas de Video en Redes P2P. Master's thesis, Facultad de Ingeniería, Montevideo, Uruguay (November 2009)
7. Yang, G., Sun, Y., Wu, H., Li, J., Liu, N., Dutkiewicz, E.: A Trust Model in P4P-integrated P2P Networks based on domain management. In: IEEE Symposium on Computers and Communications (ISCC) (June 2010)
8. Guo, Z., Min, L., Yang, H., Yang, S.: An Enhanced P4P-Based Pastry Routing Algorithm for P2P Network (August 2010)
9. Guo, Z., Yang, H., Yang, S.: P4P Pastry: A novel P4P-based Pastry routing algorithm in peer to peer network. In: The 2nd IEEE International Conference on Information Management and Engineering (ICIME 2010) (April 2010)
10. Resende, M.G.C., Ribeiro, C.C.: Greedy Randomized Adaptive Search Procedures. In: Glover, F., Kochenberger, G. (eds.) Handbook of Methaheuristics, Kluwer Academic Publishers, Dordrecht (2003)
11. Bertinat, M.E., Vera, D.D., Padula, D., Robledo, F., Rodríguez-Bocca, P., Romero, P., Rubino, G.: Goalbit: The first free and open source peer-to-peer streaming network. In: Proceedings of the 5th International IFIP/ACM Latin American Conference on Networking, pp. 49–59. ACM, New York (2009)
12. Padula, D.: Compromiso entre Pares e ISPs en el contexto P4P: Optimización en dos niveles. Master's thesis, Facultad de Ingeniería, Montevideo, Uruguay (July 2010)
13. Dale, C., Liu, J.: A measurement study of piece population in bittorrent. In: IEEE Global Telecommunications Conference, GLOBECOM, pp. 405–410. IEEE, New York (2007)

Modelling Non-stationarities in EEG Data with Robust Principal Component Analysis

Javier Pascual[1], Motoaki Kawanabe[2,1], and Carmen Vidaurre[1]

[1] Berlin Institute of Technology, Machine Learning Laboratory, Germany
[2] Fraunhofer Institute FIRST, Germany

Abstract. Modelling non-stationarities is an ubiquitous problem in neuroscience. Robust models help understand the underlying cause of the change observed in neuroscientific signals to bring new insights of brain functioning. A common neuroscientific signal to study the behaviour of the brain is electro-encephalography (EEG) because it is little intrusive, relatively cheap and easy to acquire. However, this signal is known to be highly non-stationary. In this paper we propose a robust method to visualize non-stationarities present in neuroscientific data. This method is unaffected by noise sources that are uninteresting to the cause of change, and therefore helps to better understand the neurological sources responsible for the observed non-stationarity. This technique exploits a robust version of the principal component analysis and we apply it as illustration to EEG data acquired using a brain-computer interface, which allows users to control an application through their brain activity. Non-stationarities in EEG cause a drop of performance during the operation of the brain-computer interface. Here we demonstrate how the proposed method can help to understand and design methods to deal with non-stationarities.

Keywords: non-stationarity, modelling, robust statistics, Principal Component Analysis, EEG, BCI.

1 Introduction

Understanding the cause of non-stationarity in neuro-scientific data is an important step to improve the knowledge of brain functioning. Robust models of non-stationarities are a powerful tool to understand the cause of change in neuroscientific signals. In this paper we present a method to robustly model and visualize non-stationarities in neuroscientific data and apply it to electro-encephalographic signals (EEG). EEG is a very common way to record brain activity because it is little intrusive and relatively cheap and easy to acquire. However, this signal is known to be highly non-stationary.

The method we propose is robust because it is unaffected by noise sources (aftefacts) that are not interesting to study the cause of the non-stationary present in the data. Thus, it allows researchers to better understand the neurological sources responsible for the observed statistical changes. This technique

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 51–58, 2011.

exploits a robust version of the principal component analysis (PCA). To illustrate the usefulness of the method, we apply it to EEG data acquired using a brain-computer interface (BCI) [1], which allows users to control an application through their brain activity. In the usual machine approach, a calibration session (i.e. without feedback) is necessary to extract meaningful features and train the corresponding classifier. Examples of EEG signals are collected where the user receives a cue to perform repeatedly a small number of e.g. motor imagery tasks [2]. A calibration of about 30 minutes is sufficient to adapt the system to the subject and start the feedback. During feedback the users are enabled to transfer information through their brain activity and control applications.

However, non-stationarities in EEG can cause a drop of performance (cf. [3–6]). One situation commonly affected by a performance drop is the step from calibration to feedback recordings. In this paper we apply our robust PCA to this problem and demonstrate how the proposed method can help to understand and design methods to deal with non-stationarities. The study is performed offline with data of 80 users from whom we can analyze and visualize the task-unrelated changes from calibration to feedback.

This paper is divided into five sections. Section 2 is a description of the experimental setup. Section 3 describes our proposed method. Then we present the results of the study in Section 4 and, finally a conclusion is outlined in Section 5.

2 Experimental Setup

Data were recorded in a one-day session from 80 healthy BCI-novices (39m, 41f; age 29.9±11.5y; 4 left-handed) in a common study performed by the Universities of Tübingen and Berlin [7]. Each data set was acquired during a single BCI session with a classical motor imagery paradigm. The participants were sitting in a comfortable chair with arms lying relaxed on armrests.

Brain activity was recorded from the scalp with multi-channel EEG amplifiers using 119 Ag/AgCl electrodes in an extended 10-20 system sampled at 1000 Hz with and band-pass filtered from 0.05 to 200 Hz.

First, the subjects performed three calibration runs (regarded as offline data later in the text) in which every 8s one of three different visual cues (arrows pointing left, right, down) indicated to the subject which type of motor imagery to perform: left/right hand or foot. Three runs with 25 trials of each motor condition were recorded and automatic variance based artifact rejection was applied on the calibration data to reject trials and channels.

Each binary class combination was analyzed using a semi-automatic procedure where a specific frequency band and time interval were found to calculate the filters associated to Common Spatial Patterns (CSP). This technique is widely employed in BCI research and a detailed introduction can be found in [8]. Then, 2 of the classes with best discriminability were selected and the subjects performed feedback (regarded as online data later in the text) with three runs of 100 trials each, although for some subjects only one or two runs were recorded.

The analyses presented in this paper are performed only in the two selected classes. The covariance matrices for PCA are computed from band-pass filtered EEG data in a subject-optimized frequency band (therefore their mean is zero).

3 Methods

The method suggested in this paper is inspired by the work described in [10]. There, the author proposed a low-dimensional description for the shift of covariance matrices where the data are recorded during a specific condition, in particular, a feedback BCI session consisting of several runs. Changes in the data are analyzed by applying PCA to the covariance matrices obtained from each of the feedback runs and it is shown that the first principal component is sufficient to explain the change in the data.

The work presented here differs from the previous in two aspects: we have robustified it using tools from robust statistics and we have extended it to the situation in which the shift can occur between different defined conditions (in particular, we will apply it to the change from calibration, condition 1, to feedback, condition 2).

The reason to extend the method to changes that can occur when the participant is immersed in different situations is quite obvious: to visualize and analyze the change that occurs between one and another. However, the reason to robustify the method might not be so clear. As PCA is computed from covariance matrices, it is very sensitive to the quality of their estimation. Noises due to sources different from the ones in study, might influence the parameters needed to calculate covariance matrices. For example, electrode noise very much affects the variance estimation, however, it is uninteresting to find neuroscienfic reasons for brain pattern changes.

Technically, our robust PCA is developed upon the theory of robust statistics. As in our previous work with robustly averaged covariance matrices [9], we replace the Euclidean error measure with an "outlier-insensitive" one for defining the desired low-dimensional approximations of observed data, which has the effect of minimizing the effect of noise. From this robust objective function, we derive an iterative algorithm which can automatically detect outlying information and which makes the PCA calculation robust (by reducing the influence of the noisy data through decrement of its weight). An illustrative example can be found in Figure 1.

In order to find prominent changes from one condition to the other, we propose to use the robust PCA centered at a prefixed matrix $\boldsymbol{\Sigma}_0$ whose optimization problem is formulated as follows:

$$\min_{\beta_1,\ldots,\beta_n \in \mathbb{R}, V \in Sym(d)} \sum_{i=1}^{n} \rho(||\boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_0 - \beta_i \boldsymbol{V}||^2), \quad s.t. \ ||\boldsymbol{V}|| = 1, \tag{1}$$

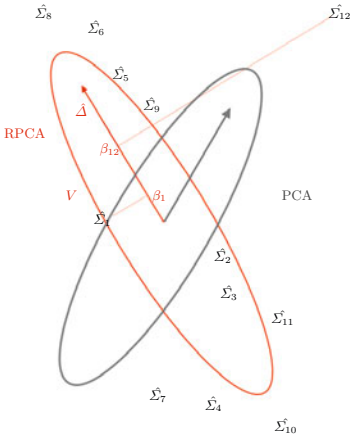


Fig. 1. The sample covariance matrices $\hat{\Sigma}_j (j = 1, ..., 12)$ are approximated linearly by calculating the largest eigen value of V $(\hat{\Delta})$, which is the covariance matrix of the vectorized $\hat{\Sigma}_j$ matrices. PCA obtains a Principal Component affected by outlying information. However, robust PCA (RPCA in the figure) minimizes the effect of $\hat{\Sigma}_{12}$ as an outlier and computes the desired direction of variance.

where $\{\boldsymbol{\Sigma}_i\}_{i=1}^n$ are sample covariance matrices under the second condition (e.g. feedback), $\boldsymbol{\Sigma}_0$ is the average covariance under the first condition (e.g. calibration), the symmetric matrix $\boldsymbol{V}$ is the principal component matrix (the most prominent change in the covariance matrices) and $\{\beta_i\}_{i=1}^n$ are the principal component scores, i.e. 1-dimensional coordinate values of the covariances $\{\boldsymbol{\Sigma}_i\}_{i=1}^n$ in the principal component direction $\boldsymbol{V}$. We deploy the standard matrix norm (Frobenius) $\|\boldsymbol{X}\|^2 = \mathrm{Tr}\left(\boldsymbol{X}\boldsymbol{X}^\top\right)$ here, and the function ρ is monotonically increasing slower than linear, which makes the principal component robust against outlying sample covariances. We remark that we obtain the ordinal matrix PCA [10], when the function ρ is linear.

The solution $\{\beta_i\}_{i=1}^n$ and $\boldsymbol{V}$ of can be obtained from the equations:

$$\beta_i \;=\; \langle \boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_0, \boldsymbol{V} \rangle\,, \tag{2}$$

$$\sum_{i=1}^{n} \rho_i' \,\langle \boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_0, \boldsymbol{V} \rangle\, (\boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_0) \;=\; \sum_{i=1}^{n} \rho_i' \,\beta_i^2 \;\boldsymbol{V}, \tag{3}$$

where $\rho_i' = \rho'\left(\|\boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_0 - \beta_i \boldsymbol{V}\|^2\right)$ and $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle := \mathrm{Tr}\left(\boldsymbol{X}\boldsymbol{Y}^\top\right)$. The first equation (2) shows that β_i is 1-dimensional projection of the difference $\boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_0$ onto the PCA direction $\boldsymbol{V}$, while the other equation (3) leads to an eigenvalue problem of larger size by vectorizing all the matrices. We remark that the weights $\{\rho_i'\}_{i=1}^n$ attenuate the influence of outlying covariances: the weight ρ_i' is small, if the 1-dimensional approximation $\boldsymbol{\Sigma}_0 + \beta_i \boldsymbol{V}$ is far from the covariance $\boldsymbol{\Sigma}_i$. By checking the weights $\{\rho_i'\}_{i=1}^n$, we can also find out the trials (time intervals) containing uninteresting noises/artefacts.

The eigen matrix $\boldsymbol{V}$ obtained by our robust PCA is the most prominent change in covariance matrices between the two conditions. However, it is not easy to interpret this matrix directly. Therefore, in our experiments, we always present the first eigen vector of the matrix $\boldsymbol{V}$, which can be nicely plotted over the scalp for further neurophysiological inspections. Our entire analysis procedure is summarized in the flow-chart Fig. 2.

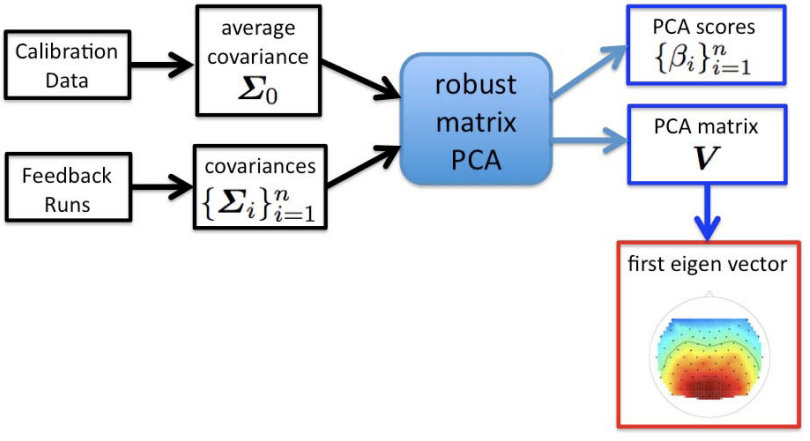


Fig. 2. The flow-chart of our robust PCA analysis for BCI data

4 Results

To illustrate the proposed method, we have chosen a classical problem in the machine learning approach to BCIs: non-stationarities become specially apparent between the initial calibration without feedback and the online operation of the system. By applying our robust PCA method, we expect to obtain a robust model that allows us to visualize the real source of change.

First, the data of the calibration and feedback runs are filtered in the frequency band obtained from the analysis of calibration data (cf. 2). Then, the data is epoched and one covariance matrix is calculated for each trial. All covariance matrices belonging to the calibration are robustly averaged to compute $\boldsymbol{\Sigma}_0$ (cf. 9). Next, the method is applied to calculate the principal source of variation between calibration and feedback.

In order to show the non-stationarity in the electrode space, we can apply an eigen-value decomposition of the matrix $\boldsymbol{V}$ (see Section 3) and choose the eigen-vector corresponding to the most extreme eigen-value. This eigen-vector can be then interpreted as the source of the main variance of the transfer from calibration to feedback conditions. Figure 3 depicts the result of applying PCA modified for two conditions (left panel) and the proposed robust PCA (right panel) to data of one BCI user. We can observe how PCA fails to show the real source of variation, whereas robust PCA minimizes the effect of outlying

and uninteresting information (an electrode artifact) and finds the real non-stationarity due to the change between calibration and feedback, which has a strong focus in the parieto-occipital area. In the analyzed data, 40% of the users (32 out of 80) suffered from some type of noise or artifact that caused the non-robust method to fail, but the actual source of change was recovered with the robust PCA version presented in this paper.

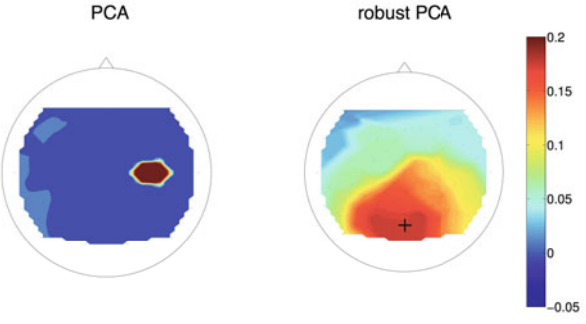


Fig. 3. Principal Component of the change from calibration to feedback for one user. The left panel shows that the result is affected by an electrode artifact and the true non-stationarity is not observable. On the right panel, the figure is obtained using the robust method. It has a strong focus in the parieto-occipital area.

The frequency band in which the feedback was performed (subject dependent, but usually either in the μ and/or β bands) is less pronounced during the online session (feedback), resulting in a strong difference between calibration and feedback conditions. This is observable in the topography displayed in Figure 3, which suggests that during the calibration a strong parietal rhythm is present due to the low visual input induced by the absence of feedback. However, that rhythmic activity is decreased during online operation due to the greater demand for visual processing. Other reasons for this effect could be considered, as for example fatigue [11], which can generate a similar topography as the one obtained here. Nevertheless, its effects are incremental over a longer time period, unlike in the step from calibration to feedback. In order to find out whether the user experienced fatigue as well, robust PCA could be applied at the beginning and end of the feedback period to observe the corresponding topography and interpret it.

In Figure 4, the spectra for the conditions offline (calibration) and online (feedback) of one channel in the parieto-occipital area is depicted, in particular the one represented with a cross in the right panel of Figure 3. The values on the horizontal color bar depict a significant difference between the two conditions when the signed squared r-value is above 0.0148 or respectively below -0.0148. The difference in power between the two spectra is obviously significant in the band of interest.

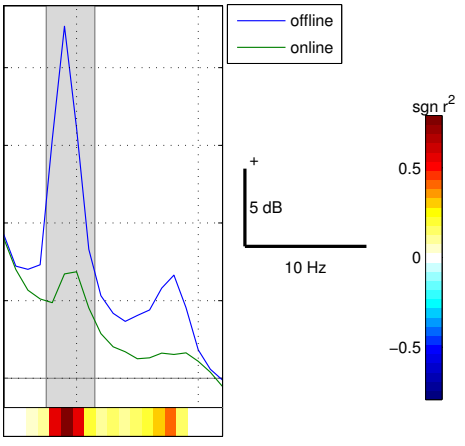


Fig. 4. Spectra for one channel (see Figure 3) in the occipital-parietal area. The power during calibration is significantly greater than the power during feedback in the band of interest (marked in gray).

5 Conclusions

In this paper, we have presented a novel algorithm to visualize the real source of neurological processes caused by non-stationarities. The method automatically reduces the effect of unrelated outlying information. We successfully applied it to model the change between calibration and feedback sessions of a BCI application with data of 80 users. For forty percent of the participants, the actual source of change was hidden behind artefactual noise but could be revealed applying our robust PCA algorithm. This result allowed to formulate a neuroscientific explanation of the now modelled and observable change. As we are able to modell the non-stationarities, we can also apply the result of the proposed analysis in other algorithms (such as CSP) to avoid, e.g., performance drops in BCI applications.

Acknowledgments. This work was supported by the grants ICT-2009-248326 and ICT-2008-216886. This publication only reflects the authors' views. Funding agencies are not liable for any use that may be made of the information contained herein.

References

1. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Braincomputer interfaces for communication and control. Clin. Neurophysiol. 113, 767–791 (2002)
2. Müller, K.R., Tangermann, M., Dornhege, G., Krauledat, M., Curio, G., Blankertz, B.: Machine learning for real-time single-trial EEG-analysis: From brain-computer interfacing to mental state monitoring. J. Neurosci. Methods 167(1), 82–90 (2008)

3. Vidaurre, C., Kawanabe, M., von Bünau, P., Blankertz, B., Müller, K.R.: Toward an unsupervised adaptation of LDA for Brain Computer Interfaces. IEEE Trans. Biomed. Eng (2011)
4. Vidaurre, C., Sannelli, C., Müller, K.R., Blankertz, B.: Machine-Learning Based Co-adaptive Calibration. Neural Comput. (in press 2011)
5. Vidaurre, C., Sannelli, C., Müller, K.R., Blankertz, B.: Co-adaptive calibration to improve BCI efficiency. J. Neural Eng (2011) (in press)
6. Vidaurre, C., Krämer, N., Blankertz, B., Schlögl, A.: Time Domain Parameters as a feature for EEG-based Brain Computer Interfaces. Neural Networks 22, 1313–1319 (2009)
7. Blankertz, B., Sannelli, C., Halder, S., Hammer, E.M., Kübler, A., Müller, K.R., Curio, G., Dickhaus, T.: Neurophysiological Predictor of SMR-Based BCI Performance. NeuroImage 51(4), 1303–1309 (2010)
8. Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., Müller, K.R.: Optimizing Spatial Filters for Robust EEG Single-Trial Analysis. IEEE Signal Process Mag. 25(1), 41–56 (2008)
9. Kawanabe, M., Vidaurre, C.: Improving BCI performance by modified common spatial patterns with robustly averaged covariance matrices. In: WC 2009, pp. 279–282 (2009)
10. Krauledat, M.: Analysis of Nonstationarities in EEG signals for improving Brain-Computer Interface performance. PhD thesis, Technische Universität Berlin, Fakultät IV – Elektrotechnik und Informatik (2008)
11. Lorist, M., Bezdan, E., ten Caat, M., Span, M., Roerdink, J., Maurits, N.: The influence of mental fatigue and motivation on neural network dynamics; an EEG coherence study

Performance Evaluation of Road Traffic Control Using a Fuzzy Cellular Model

Bartłomiej Płaczek

Faculty of Transport, Silesian University of Technology,
Krasińskiego 8, 40-019 Katowice, Poland
bartlomiej.placzek@polsl.pl

Abstract. In this paper a method is proposed for performance evaluation of road traffic control strategies. The method is designed to be implemented in an on-line simulation environment, which enables optimisation of adaptive traffic control. Performance measures are computed using a fuzzy cellular traffic model, formulated as a hybrid system combining cellular automata and fuzzy calculus. Experimental results show that the introduced method allows the performance to be evaluated using imprecise traffic data. The fuzzy definitions of performance measures are convenient for uncertainty determination in traffic control decisions.

Keywords: fuzzy numbers, cellular automata, road traffic control.

1 Introduction

On-line simulation is an effective approach for performance evaluation and optimisation of adaptive road traffic control [5]. The term on-line means that the simulation is synchronised with real time and it is adjusted to traffic data. In this technique real-time traffic measurements are the main inputs to a traffic model, which is used to predict future evolution of the traffic flow. Applying a predictive traffic model, the performance of traffic control is estimated in terms of travel times, delays, queue lengths, etc. These performance measures and their predicted short-term evolution are compared for all alternative control strategies (e.g. travel routes or traffic signal timings). On this basis, a strategy that leads to an optimal performance of the traffic control is selected for implementation.

An important requisite for the on-line simulation is a suitable traffic model that should present a well-balanced trade-off between accuracy and computational complexity to enable the on-line processing of traffic data. A representation of uncertainty is necessary in the model to take into account the inherent complexity of traffic processes. Furthermore, the individual features of vehicles have to be modelled as they have a significant influence on the performance of traffic control. The traffic model also has to provide interfaces for many different sources of traffic data that have become available recently (e.g. vision-based monitoring systems [11] and vehicular sensor networks [7]). To facilitate the on-line simulation, the traffic model has to be adjusted in order to maintain consistency between simulated and measured traffic.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 59–66, 2011.

The issues discussed above have motivated the development of a fuzzy cellular traffic model, which was formulated as a hybrid system combining cellular automata and fuzzy calculus [10]. The fuzzy cellular model is based on a cellular automata approach to traffic modelling that ensures the accurate simulation of real traffic phenomena (for an overview see [8]). The original feature distinguishing this model from the other cellular models is that vehicle position, its velocity and other parameters are modelled by fuzzy numbers. Moreover, the rule of model transition from one time step to the next is also based on fuzzy definitions of basic arithmetic operations. The application of fuzzy calculus helps to deal with imprecise traffic data and describe uncertainty of the simulation results. These facts along with low computational complexity make the model suitable for the on-line processing of traffic data.

Hybrid systems that combine the cellular automata and fuzzy logic are typically referred to as fuzzy cellular automata (FCA) [2]. FCA-based models have found many applications in the field of complex systems simulation e.g. [1], [12]. A road traffic model of this kind has been proposed in [3]. In such models, the local update rule of classical cellular automata is usually replaced by a fuzzy logic system consisting of fuzzy rules, fuzzification, inference, and defuzzification mechanisms. A different approach is used in this paper: current states of the cells are determined by fuzzy sets and calculus with fuzzy numbers is involved in the update operation. The innovative features of the proposed methodology are the elimination of information loss caused by defuzzification and the incorporation of uncertainty in simulation results.

In this paper the fuzzy cellular model is applied to on-line simulation in order to evaluate the performance of road traffic control. To reduce the computational effort associated with on-line simulation, the fuzzy cellular model is implemented using a concept of ordered fuzzy numbers [4]. Algebra of the ordered fuzzy numbers is significantly more efficient than the solution based on classical fuzzy numbers and extension principle applied in [10]. These advantages enable computationally efficient evaluation of performance measures that are represented by means of fuzzy numbers. As it is shown in this paper, this representation is convenient for the determination of uncertainty in traffic control decisions.

The introduced method provides significant improvement when comparing with the state-of-the-art techniques. Existing methods use traffic parameters that describe queues or groups of vehicles rather than individual cars [6], [14]. However, modern sensing platforms offer traffic data concerning the parameters of particular vehicles (position, velocity, acceleration, class, etc.) [7]. These data cannot be fully utilised when using the existing methods. In the proposed approach the precision of traffic state description is variable and can be adjusted to match the precision of available traffic data.

The rest of the paper is organised as follows: Section 2 describes the fuzzy cellular traffic model. Basic measures for the evaluation of traffic control performance are defined in Section 3. In Section 4, an issue of imprecise traffic data processing is discussed. Section 5 contains the results of an experimental study. Finally, conclusions are given in Section 6.

2 Traffic Model for On-Line Data Processing

A traffic lane in the fuzzy cellular model is divided into cells that correspond to the road segments of equal length. The traffic state is described in discrete time steps. These two basic assumptions are consistent with those of the Nagel-Schreckenberg cellular automata model. Thus, the calibration methods proposed in [9] are also applicable here for the determination of cells length and vehicles properties. A novel feature in this approach is that vehicle parameters are modelled using ordered fuzzy numbers [4]. The rule of model transition from one time step to the next is also based on fuzzy definitions of basic arithmetic operations.

Hereinafter, all the ordered fuzzy numbers are represented by four integers and the following notation is used: $A = (a^{(1)}, a^{(2)}, a^{(3)}, a^{(4)})$. This notation is suitable for both triangular as well as trapezoidal membership functions. The arithmetic operations of addition and subtraction as well as the minimum function are computed for the ordered fuzzy numbers A, B using the following definition:

$$o(A, B) = (o(a^{(1)}, b^{(1)}), o(a^{(2)}, b^{(2)}), o(a^{(3)}, b^{(3)}), o(a^{(4)}, b^{(4)})), \tag{1}$$

where o stands for an arbitrary binary operation.

The road traffic stream is represented in the fuzzy cellular model as a set of vehicles. A vehicle n is described by its position $X_{n,t}$, velocity $V_{n,t}$ (in cells per time step), maximal velocity $V_{\max,n}$ and acceleration A_n. All these quantities are expressed by fuzzy numbers. The position $X_{n,t}$ is a fuzzy number defined on the set of cells indexes. Velocity of vehicle n at time step t is computed as follows:

$$V_{n,t} = \min\left\{V_{n,t-1} + A_n(V_{n,t-1}), G_{n,t}, V_{\max,n}\right\}, \tag{2}$$

where $G_{n,t} = X_{n-1,t} - X_{n,t} - (1,1,1,1)$ is the fuzzy number of free cells in front of a vehicle n, $n-1$ denotes the number corresponding to the lead vehicle. If there is no lead vehicle in front of the vehicle n then $G_{n,t}$ is assumed to be equal to $V_{max,n}$. Acceleration is defined as a function of velocity to enable implementation of a slow-to-stop rule that exhibits more realistic driver behaviour [8]. After determination of velocities for all vehicles, their positions are updated as follows:

$$X_{n,t+1} = X_{n,t} + V_{n,t}. \tag{3}$$

The preceding formulation of the traffic model is illustrated in Fig. 1, which shows the results of numerical motion simulation of two accelerating vehicles during three time steps. Fig. 1 presents membership functions of the fuzzy numbers representing vehicles positions, the remaining parameters of vehicles are listed in Table 1. The maximal velocity of vehicles in this example is set as follows: $V_{max,n} = (1,2,2,3)$. The acceleration can take two fuzzy values depending on the vehicles velocity: $A_n(V_{n,t-1}) = (1,1,1,1)$ if $V_{n,t-1} = V_{max,n}$ or $V_{n,t-1} = V_{max,n} - (1,0,0,0)$ and $A_n(V_{n,t-1}) = (0,1,1,1)$ else.

Since the fuzzy cellular model is designed to be used for the performance evaluation of traffic control, it has to take into account the status of traffic control operations. In case of a traffic signal control, drivers reactions to traffic

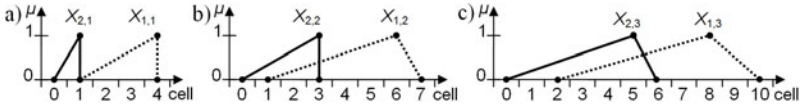


Fig. 1. Motion simulation of two vehicles: a) $t = 1$, b) $t = 2$ c) $t = 3$

Table 1. Vehicles parameters for the simulation illustrated in Fig. 1

t	$X_{1,t}$	$A_{1,t}$	$G_{1,t}$	$V_{1,t}$	$X_{2,t}$	$A_{2,t}$	$G_{2,t}$	$V_{2,t}$
0	(1,2,2,2)	(0,1,1,1)	$V_{\max,1}$	(0,2,2,2)	(0,0,0,0)	(0,1,1,1)	(0,1,1,1)	(0,1,1,1)
1	(1,4,4,4)	(0,1,1,1)	$V_{\max,1}$	(0,2,2,3)	(0,1,1,1)	(0,1,1,1)	(0,2,2,2)	(0,2,2,2)
2	(1,6,6,7)	(1,1,1,1)	$V_{\max,1}$	(1,2,2,3)	(0,3,3,3)	(0,1,1,1)	(0,2,2,3)	(0,2,2,3)
3	(2,8,8,10)	(1,1,1,1)	$V_{\max,1}$	(1,2,2,3)	(0,5,5,6)	(1,1,1,1)	(1,3,3,3)	(1,2,2,3)

signals need to be considered. The influence of traffic signalisation is simulated in this study by introducing phantom vehicles in cells corresponding with the locations of traffic signals. Actual as well as maximal velocity of a phantom vehicle is always equal zero $(0, 0, 0, 0)$. Such vehicle is inserted into simulation during the red signal period and removed when the green signal is active.

3 Performance Measures

There are several different measures available that can be employed for the evaluation of traffic control performance e.g.: average delay per vehicle, maximum individual delay, percentage of cars that are stopped, average number of stops, queue length, throughput of intersections, and travel time. In this section an algorithm is provided for computing the basic performance measures in the fuzzy cellular model. For the formal presentation of the algorithm a function S_C is defined that acts on directed fuzzy numbers:

$$S_C(A) = (s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}),\ s^{(i)} = \begin{cases} 0, |A_C| < 5 - i \\ 1, |A_C| \geq 5 - i \end{cases} \tag{4}$$

where A_C is a set of integers used in notation of the fuzzy number A, which satisfy condition denoted by C: $A_C = \left\{a^{(i)} \in \left\{a^{(1)}, a^{(2)}, a^{(3)}, a^{(4)}\right\} : a^{(i)} \text{ satisfies } C\right\}$.

Function $S_C(A)$ allows us to determine a level of confidence that the condition C is satisfied by A. The value of this function is $(0, 0, 0, 0)$ if the condition C is false for A and $(1, 1, 1, 1)$ if the condition is true. Any other combination of $s^{(i)}$ means that the condition is partially satisfied by A. For example, let us check the condition "vehicle is stopped" for the second time step of the simulation presented in Fig. 1 ($t = 2$). Such a condition can be written as $V_{n,2} = 0$ and the corresponding form of the function (4) is $S_{=0}(V_{n,2})$. The values of this function are as follows: $S_{=0}(V_{1,2}) = (0, 0, 0, 0)$ for the lead vehicle and $S_{=0}(V_{2,2}) = (0, 0, 0, 1)$ for the following vehicle (see Table 1). It means that the lead vehicle is not stopped and there is a possibility that the following vehicle is stopped.

Traffic performance measures can be now formulated in terms of the fuzzy cellular model, using the function S_C. The most commonly used criteria of performance are average delay and average number of stops. Average delay in time steps per vehicle is defined as:

$$D = \frac{1}{N} \sum_n \sum_t S_{=0}(V_{n,t}) \tag{5}$$

where N is the total number of vehicles. The quotient A/N of an ordered fuzzy number A and an integer N is computed according to formula: $A/N = (a^{(1)}/N, a^{(2)}/N, a^{(3)}/N, a^{(4)}/N)$. Additionally, the values of $a^{(i)}/N$ are rounded to the nearest integers.

A stop of a vehicle is the situation in which the velocity of a vehicle is higher than zero for the previous time step of simulation $(t-1)$ and it is zero for the current time step (t). Thus, the average number of stops is given by:

$$S = \frac{1}{N} \sum_n \sum_t \min \{S_{>0}(V_{n,t-1}), S_{=0}(V_{n,t})\} \tag{6}$$

If a vehicle n is stopped in queue, then the number $G_{n,t}$ of free cells in front of it is zero. In this study the length of a vehicle is assumed to be equal one cell, therefore the average queue length in cells can be computed using the following formula:

$$Q = \frac{1}{T} \sum_n \sum_t S_{=0}(G_{n,t}) \tag{7}$$

where T is the number of steps in the analysed time period of traffic simulation.

The main advantage of the presented model relies on the fact that the performance estimation of traffic control is computationally efficient and the uncertainty of the results is taken into account. The results concerning traffic performance are represented by means of fuzzy numbers. As it is shown in Section 5, this representation is convenient for the determination of uncertainty in control decisions.

4 Modelling of an Imprecise Traffic Information

Precise data on the parameters of each particular vehicle are often hardly available due to lack of sensing devices and high transmission costs. This section discusses an issue of imprecise traffic data processing with application of the fuzzy cellular model.

Let us assume that the data on vehicles locations are delivered with a precision, which is insufficient to place vehicles in specific cells. It means that the available traffic information describes the number of vehicles that are located in a given segment of cells. In Fig. 2 an example is illustrated, which corresponds to information indicating that there are three vehicles present in the segment of fifteen cells. This information is introduced in the model through assumption of the four instances that are shown in Fig. 2 a)-d). For the instances a and b it was assumed that the vehicles are equally spaced and are moving with maximal velocity. The instances c and d represent the least possible situations in which

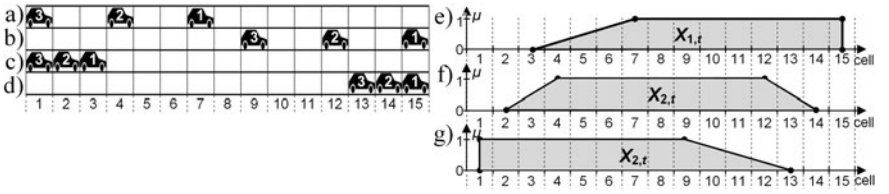


Fig. 2. Representation of imprecise traffic information in fuzzy cellular model

the vehicles are stopped in a queue. A fusion of these four instances is achieved by using ordered fuzzy numbers to describe the vehicles positions (Fig. 2 e-g).

To generalize the example discussed above, let us consider N vehicles located in a road segment that consists of cells $c_S, ..., c_E$. A position of vehicle n ($n = 1, ..., N$) is defined by the following equations:

$$x_{n,t}^{(1)} = c_S + N - n,\ x_{n,t}^{(2)} = c_S + \sum_{i=n+1}^{N} \min\left\{\lfloor L/N \rfloor, v_{\max,i}^{(2)} + 1\right\},$$
$$x_{n,t}^{(3)} = c_E - \sum_{i=2}^{n} \min\left\{\lfloor L/N \rfloor, v_{\max,i}^{(3)} + 1\right\},\ x_{n,t}^{(4)} = c_E + 1 - n, \qquad (8)$$

where $L = c_S - c_E + 1$ is the number of cells in the road segment (precision unit).

The above definition allows the imprecise data to be utilised for the evaluation of traffic control performance. It demonstrates also the applicability of the introduced approach for modelling the imprecise traffic information.

5 Case Study

The fuzzy cellular model was applied to performance evaluation of traffic control at a road work zone. Fig. 3 a) shows a schematic layout of the simulated situation. Average delay was analysed in this experiment for two different control strategies. At the beginning of each simulation vehicles were randomly placed on road lanes approaching the work site (lanes A and B). In the first strategy a green signal is displayed for the traffic lane A. Afterwards, when all vehicles vacate this lane, the green light is activated for the opposite direction (lane B). The second control strategy assumes that an inverse traffic signal sequence is used (the green signal for lane B is activated at first). For both strategies the average delay per vehicle was calculated as a fuzzy number according to equation (5). Simulations were executed for various numbers of vehicles generated in the traffic lanes and for different precision units. The precision unit L was defined as the number of cells in segments that are used for determination of the initial vehicles locations.

The average delays $D_i = (d_i^{(1)}, d_i^{(2)}, d_i^{(3)}, d_i^{(4)})$ evaluated for both control strategies ($i = 1, 2$) have to be compared in order to select the optimal strategy for implementation. An example of such comparison is presented in Fig. 3 b) and c). The simulation was executed for 50 and 45 vehicles placed in lanes A and B respectively ($N_A = 50, N_B = 45$). It can be observed that the optimal control strategy cannot be selected without ambiguity and the uncertainty of this

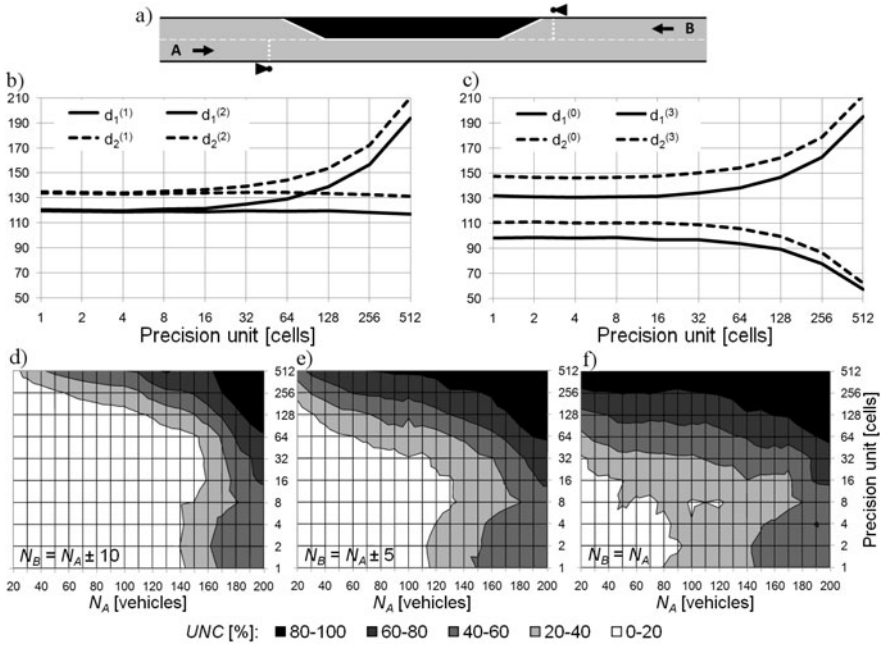


Fig. 3. a) Lane closure on two-lane road using traffic signals, b)-f) Experimental results

selection arises when higher precision units are in use. The uncertainty of strategy selection was determined using the following formula:

$$UNC = 1 - \max\{P(D_1 < D_2), P(D_2 < D_1)\} + \min\{P(D_1 < D_2), P(D_2 < D_1)\}, \quad (9)$$

where $P(C)$ denotes a probability of satisfying condition C. The probabilities in equation (9) were computed using a method of fuzzy numbers comparison proposed in [13]. The uncertainty takes a value between 0 and 1 (0 - 100 [%]). Contour plots in Fig. 3 d)-f) present the results of uncertainty determination that were obtained assuming various absolute differences between the numbers of vehicles placed in traffic lanes A and B ($|N_A - N_B|$). The differences are equal 10, 5 and 0 for Fig. 3 d), e) and f) respectively. It can be seen that the uncertainty increases when the difference $|N_A - N_B|$ is decreased. The uncertainty level is similar for precision units in a range from 1 to 16 cells; however, further decrease in the traffic data precision causes a significant increase of the uncertainty. This way of analysis allows us to determine the minimal precision of traffic data, which is necessary to select the optimal control strategy using the proposed performance evaluation method.

6 Conclusions

The fuzzy cellular model enables the evaluation of performance measures for road traffic control. Imprecise traffic data can be utilised in the proposed hybrid

modelling approach. The results of performance evaluation are represented in terms of ordered fuzzy numbers. This representation is suitable for the uncertainty determination in traffic control decisions that are based on the performance comparison for several candidate control strategies. The uncertainty level can be used to decide if the currently available traffic data are sufficient for the selection of optimal control strategy. It allows the precision level of traffic measurements to be dynamically adapted to both the current traffic situation and the requirements of traffic control system.

References

1. Bone, C., Dragicevic, S., Roberts, A.: A fuzzy-constrained cellular automata model of forest insect infestations. Ecol. Model. 192, 107–125 (2006)
2. Chang, C., Zhang, Y., Dong, Y.: Application of fuzzy logic in the classical cellular automata model. J. Appl. Math. & Computing 20(12), 433–443 (2006)
3. Gong, Y.G., Liu, L.: A Fuzzy Cellular Automaton Model Based On NaSchH Model. In: 2nd Int. Conf. on Signal Processing Systems, pp. V2-518–V2-522. IEEE, Los Alamitos (2010)
4. Kosiński, W.: On Fuzzy Number Calculus and Some Application. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 250–259. Springer, Heidelberg (2006)
5. Kosonen, I.: Multi-agent fuzzy signal control based on real-time simulation. Transport. Res. C-Emer. 11(5), 389–403 (2003)
6. Kurzhanskiy, A.A., Varaiya, P.: Active traffic management on road networks: a macroscopic approach. Phil. Trans. R. Soc. A 368, 4607–4626 (2010)
7. Lee, U., Gerla, M.: A survey of urban vehicular sensing platforms. Comput. Netw. 54(4), 527–544 (2010)
8. Maerivoet, S., De Moor, B.: Cellular automata models of road traffic. Phys. Rep. 419, 1–64 (2005)
9. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. J. Phys. I France 2(12), 2221–2229 (1992)
10. Płaczek, B.: Fuzzy Cellular Model for On-Line Traffic Simulation. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2009. LNCS, vol. 6068, pp. 553–560. Springer, Heidelberg (2010)
11. Płaczek, B., Staniek, M.: Model Based Vehicle Extraction and Tracking for Road Traffic Control. In: Kurzyński, M., et al. (eds.) Computer Recognition Systems 2. ASC, vol. 45, pp. 844–851. Springer, Heidelberg (2007)
12. Ramirez, C.L., Castillo, O.: A Hybrid Model Based on a Cellular Automata and Fuzzy Logic to Simulate the Population Dynamics. In: Castillo, O., et al. (eds.) Soft Computing for Hybrid Intel. Systems, pp. 189–203. Springer, Heidelberg (2008)
13. Sevastianov, P.: Numerical methods for interval and fuzzy number comparison based on the probabilistic approach and Dempster-Shafer theory. Inform. Sciences 177(21), 4645–4661 (2007)
14. Van den Berg, M., Hegyi, A., De Schutter, B., Hellendoorn, J.: A macroscopic traffic flow model for integrated control of freeway and urban traffic networks. In: Proc. of the 42nd IEEE Conf. on Decision and Control, pp. 2774–2779 (2003)

An Improved Heuristic for the Bandwidth Minimization Based on Genetic Programming

P.C. Pop[1] and O. Matei[2]

[1] Dept. of Mathematics and Informatics, North University of Baia Mare, Romania
petrica.pop@ubm.ro
[2] Dept. of Electrical Engineering, North University of Baia Mare, Romania
oliviu.matei@holisun.com

Abstract. In this work we develop an improved heuristic based on genetic programming (GP) for the matrix bandwidth minimization problem (MBMP). This problem consists in rearranging the rows and columns of a sparse matrix such that the non-zero elements are in a band as close as possible to the main diagonal. We evaluated our heuristic on a set of 25 benchmark instances from the literature and compared with state-of-the-art algorithms. The obtained results are very encouraging and point out that GP is an appropriate method for solving the MBMP.

Keywords: bandwidth minimization problem, heuristics, genetic programming.

1 Introduction

The matrix bandwidth minimization problem (MBMP) has been subject of research for at least 42 years, beginning with the Cuthill-McKee algorithm in 1969. The problem consists of finding the a permutation of the rows and columns of a matrix that keeps all the non-zero elements in a band that is as close as possible to the main diagonal.

Formally, the MBMP can be stated as follows: given a sparse matrix $A = [a_{ij}]_{n\times n}$, we are interested in finding a permutation of the rows and columns that minimizes the distance b of any non-zero entry from the center diagonal. The problem plays an important role in solving large linear systems because Gaussian elimination can be performed in $O(nb^2)$ on matrices of bandwidth b, while in the general case the algorithm is performed in $O(n^3)$.

The problem can be defined in the context of graphs as follows: let $G = (V, E)$ be a finite graph with $|V| = n$ and $f : V \rightarrow \{1, ..., n\}$ a labelling of its nodes, then the *bandwidth* of G under f is defined as:

$$B_f(G) = max\{|f(v_i) - f(v_j)| \ : \ (v_i, v_j) \in E\}. \tag{1}$$

Then the bandwidth minimization problem (BMP) consists of finding a labeling f which minimizes $B_f(G)$.

The graph and the matrix versions of the bandwidth problem are equivalent. The equivalence is clear if we replace the nonzero entries of the matrix by 1's and

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 67–74, 2011.

interpret the result as the adjacency matrix of a graph. In [10], it is described an example that shows this equivalence.

On graphs the BMP arises on more subtle ways and finds several applications: circuit design, data mining, VLSI design, network survivability, data storage, industrial electromagnetics, etc.

The bandwidth problem is NP-hard [11], even for some special cases and as well is NP-hard to approximate within any constant.

Due to its practical applications, the MBMP has generated a considerable interest over the years being proposed several exact and heuristic algorithms. Corso and Manzini [1] proposed two exact algorithms that solved problems for randomly generated graphs with nodes up to 100. The first heuristic algorithm for solving the MBMP was proposed by Cuthill-McKee [2]. The difficulty of obtaining optimum solutions for the MBMP has led to the development of several metaheuristics. The first such algorithms were the Tabu Search (TS) heuristic proposed by Marti *et al.* [8], the Greedy Randomized Adaptive Search Procedure (GRASP) combined with a Path Relinking (PR) method described by Pinana *et al.* [12] and the genetic algorithm (GA) suggested by Lim *et al.* [5]. More recently, Rodrigues-Tello *et al.* [13] proposed an improved Simulated Annealing (SA) heuristic, Koohestani and Poli [3] described the first genetic programming approach and finally, Mladenovic *et al.* [10] elaborated a variable neighbourhood search (VNS) based heuristic for reducing the bandwidth of a matrix.

The aim of this paper is to describe an improved heuristic based on genetic programming for solving the MBMP. Our heuristic is tested against state-of-the-art algorithms on a set of 25 benchmark instances from the literature and as will be shown in the computational experiments section, the proposed approach provides high quality solutions.

2 The Genetic Program for Solving the MBMP

Genetic programming addresses one of the central goals of computer science, namely automated programming, whose goal is to create, in an automated way, a computer program that enables the computer to solve the problem. Koza [4] suggested that the desired program should evolve itself during the evolution process. In other words, instead of solving a problem by building an evolution program that solves it, we should rather search the space of possible computer programs for the best one. This evolution method is called Genetic Programming (GP).

Genetic programming is a branch of genetic algorithms. The main difference between genetic programming and genetic algorithms is the representation of the solution, namely, genetic programming creates computer programs in the Lisp or scheme computer languages as the solution while genetic algorithms create a string of numbers that represent the solution.

In genetic programming are used four steps in order to solve problems:

1) Generate an initial population of random compositions of the functions and terminals of the problem (computer programs).

2) Execute each program in the population and assign it a fitness value according to how well it solves the problem.
3) Create a new population of computer programs.
   i) Copy the best existing programs
   ii) Create new computer programs by mutation.
   iii) Create new computer programs by crossover.
4) The best computer program that appeared in any generation, the best-so-far solution, is designated as the result of genetic programming.

In what it follows we present an heuristic algorithm for solving the MBMP based on genetic programming.

2.1 Genetic Representation

An individual is represented as a list of interchanges of lines or columns:

$$I = (W_1 < k_s^1, k_d^1 >, W_2 < k_s^2, k_d^2 >, ..., W_m < k_s^m, k_d^m >), \tag{2}$$

where $W_i \in {}'L', 'C'$ ('L' means an interchange of lines, and 'C' is an interchange of columns) and k_s^i respectively k_d^i are the two lines/columns to be interchanged. The permutations are applied successively, in order. Given the following matrix:

$$A = \begin{pmatrix} 11\ 12\ 13 \\ 21\ 22\ 23 \\ 31\ 32\ 33 \end{pmatrix} \tag{3}$$

and the individual $I = (C < 1, 3 >, L < 2, 1 >)$, the matrix undergoes the following permutations:

1. the columns 1 and 3 are interchanged and results the following matrix:

$$A_1 = \begin{pmatrix} 13\ 12\ 11 \\ 23\ 22\ 21 \\ 23\ 32\ 31 \end{pmatrix}, \tag{4}$$

2. the lines 2 and 1 are interchanged and the result is:

$$A_2 = \begin{pmatrix} 23\ 22\ 21 \\ 13\ 12\ 11 \\ 23\ 32\ 31 \end{pmatrix}. \tag{5}$$

Therefore the resulted matrix after applying the individual (program) I is A_2.

2.2 Initial Population

In our program the initial population is generated randomly. The length of each individual is chosen at random, up to twice the size of the matrix.

2.3 The Fitness Value

Every solution has a fitness value assigned to it, which measures its quality. In our case, the fitness value is given by the bandwidth of the matrix resulted after a program is applied to the original matrix.

2.4 Genetic Operators

We considered three operations for modifying structures in genetic programming: crossover, mutation and pruning. The most important one is the crossover operation. In the crossover operation, two solutions are combined to form two new solutions or offspring.

Crossover. Two parents are selected from the population by the binary tournament method. The two programs can undergo two types of crossover with the same probability: one-cut crossover, respectively a concatenation.

- One-cut crossover operator
  Offspring are produced from two parent solutions using the following crossover procedure described by Matei in [9]: it creates offspring which preserve the order and position of symbols in a subsequence of one parent while preserving the relative order of the remaining symbols from the other parent. It is implemented by selecting a random cut point. The first offspring is made of the first part of the first parent, respectively the second part of the second parent. The other offspring is made of the second sequence of the first parent, respectively the first sequence of the first parent.
  Given the two parents:

$$P_1 = (M_1^1 M_2^1 | M_3^1 M_4^1), \tag{6}$$
$$P_2 = (M_1^2 M_2^2 | M_3^2 M_4^2 M_5^2), \tag{7}$$

  where the superior index represents the parent (first or second), the number of elements of the parent represent the number of permutations (interchanges of lines or columns) and "|" defines the cutting point, then the offspring are:

$$O_1 = (M_1^1 M_2^1 | M_3^2 M_4^2 M_5^2), \tag{8}$$
$$O_2 = (M_1^2 M_2^2 | M_3^1 M_4^1). \tag{9}$$

- Concatenation operator
  The concatenation operator concatenates two programs. The first offspring is formed by adding the second parent at the end of the first one. The other offspring is made of the second parent followed by the first one.
  Given the same two parents:

$$P_1 = (M_1^1 M_2^1 M_3^1 M_4^1), \tag{10}$$
$$P_2 = (M_1^2 M_2^2 M_3^2 M_4^2 M_5^2), \tag{11}$$

the offspring are:

$$P_1 = (M_1^1 M_2^1 M_3^1 M_4^1 | M_1^2 M_2^2 M_3^2 M_4^2 M_5^2), \tag{12}$$

$$P_2 = (M_1^2 M_2^2 M_3^2 M_4^2 M_5^2 | M_1^1 M_2^1 M_3^1 M_4^1). \tag{13}$$

This example shows one of the main advantages of genetic programming over genetic algorithms: in genetic programming identical parents can yield different offspring, while in genetic algorithms identical parents would yield identical offspring.

The concatenation operator is of great importance because it is the most important way of evolving programs to longer ones (with more moves).

Mutation. Mutation is another important feature of genetic programming. We use in our GP six random mutation operators chosen with the same probability:

1. addition of a move: an allele (position in an individual) is chosen at random and a new random move in inserted in that place. This way, the length of the individual increases.
2. removal of a move: a move is chosen randomly and removed from the individual. This way, the length of the individual decreases.
3. exchange of two moves: two alleles randomly selected are swapped.
4. replacement of the item which undergoes the move:] a column is replaced by a line or the other way around: e.g. $L < 1, 4 >$ is replaced by $C < 1, 4 >$.
5. replacement of the position of the items which undergo the move: e.g. $L < 2, 3 >$ by $L < 1, 5 >$.
6. replacement of the entire move: a randomly selected allele is replaced by a new one, yet generated randomly.

The choice of which of the operators described above should be used to create an offspring is probabilistic. Their probability of applications are called operator rates. Typically, crossover is applied with highest probability, the crossover rate being 90% or higher. On the contrary, the mutation rate is much smaller, typically being in the region of 10%.

Pruning. It is often the case that longer individuals contain shorter sequences of moves which lead to better results. This is the reason for introducing a new operator, called *pruning*.

Given an individual:

$$P = M_1 M_2 M_3 ... M_k M_{k+1} ... M_n, \tag{14}$$

the pruning operator generates a new individual:

$$P = M_1 M_2 M_3 ... M_k \tag{15}$$

holding the following condition:

$$f([M_1M_2M_3...M_k]) = \min_{i=\overrightarrow{1,n}} f([M_1...M_i]), \tag{16}$$

where $f([M_1...M_i])$ is the fitness value of the sequence of moves $[M_1...M_i]$.

The pruning operator is deterministic and it is applied for all individuals.

2.5 Selection

The selection process is deterministic. In our algorithm we use the $(\mu + \lambda)$ selection, where μ parents produce λ offspring. The new population of $(\mu + \lambda)$ is reduced again to μ individuals by a selection based of the "survival of the fittest" principle. In other words, parents survive until they are suppressed by better offspring. It might be possible for very well adapted individuals to survive forever.

2.6 Genetic Parameters

The genetic parameters are very important for the success of a GP, equally important as the other aspects, such as the representation of the individuals, the initial population and the genetic operators. The most important parameters are:

- the population size μ has been set to 5 times the size of the matrix. This turned out to be the best number of individuals in a generation.
- the intermediate population size λ was chosen twice the size of the population: $\lambda = 2 \cdot \mu$.
- mutation probability was set at 10%.

In our algorithm the termination strategy is based on a maximum number of generations to be run.

3 Computational Results

The performance of our heuristic approach based on GP was tested on 25 benchmark instances from the Harwell-Boeing sparse matrix collection (available from http://math.nist.gov/MatrixMarket/data/Harwell-Boeing). The Harwell-Boeing Sparse Matrix Collection is a set of standard test matrices arising from problems in linear systems, least squares, and eigenvalue calculations from a wide variety of scientific and engineering disciplines.

The testing machine was an Intel Dual-Core 1,6 GHz and 1 GB RAM. The operating system was Windows XP Professional. The algorithm was developed in Java, JDK 1.6.

In the next table we compared the solution qualities (i.e. the minimum bandwidth obtained) of our genetic programming based heuristic with other five best

Table 1. Computational results for the 25 medium instances from the Harwell-Boeing Sparse Matrix Collection

Instance	n	GP	VNS [10]	SA [13]	GA [5]	GRASP-PR [12]	TS [7]
arc130	130	63	63	63	63	63	64
ash85	85	9	9	9	9	9	9
bcspwr01	39	5	5	5	5	5	5
bcspwr02	49	7	7	7	7	7	7
bcspwr03	118	**9**	10	10	12	11	11
bcsstk01	119	16	16	16	16	16	17
can_144	144	13	13	13	15	14	14
can_161	161	18	18	18	19	18	19
dwt_234	117	11	12	11	12	11	11
gent113	104	**25**	27	27	27	27	27
gre_115	115	**22**	23	23	24	24	24
gre_185	185	**19**	21	22	22	22	22
ibm32	132	11	11	11	11	11	12
impcol_b	59	**19**	20	20	20	21	21
impcol_c	137	**27**	30	30	31	31	32
lund_a	147	23	23	23	23	23	23
lund_b	147	23	23	23	23	23	23
mcca	168	**32**	37	37	37	37	37
nos1	158	3	3	3	4	3	3
nos4	100	10	10	10	10	10	10
steam3	80	7	7	7	7	7	7
west0132	132	**28**	32	33	33	35	34
west0156	156	**33**	36	36	37	37	37
west0167	167	34	34	34	36	35	34
will199	199	66	65	65	66	69	67

algorithms from the literature: Variable Neighborhood Search (VNS) [10], Simulated Annealing (SA) [13], Genetic Algorithm (GA) [5], Greedy Randomized Adaptive Search Procedure Path Relinking (GRASP-PR) [12] and Tabu Search (TS) [8].

Analyzing the computational results, it results that our proposed heuristic based on GP performs very well in terms of solution quality in comparison with the state-of-the-art algorithms for solving the MBMP: in 15 out of 25 instances we obtained the same bandwidth as the best of the compared algorithms, in 9 out of 25 instances we improved the value of the bandwidth and in one case (instance will199) the bandwidth provided is higher than solution provided by the VNS and SA algorithms.

Regarding the computational times, it is difficult to make a fair comparison between algorithms, because they have been evaluated on different computers and they are implemented in different languages. However, it should be noted that our heuristic is slower than the compared algorithms and therefore our approach will be appropriate when the execution speed is not critical.

4 Conclusions

We described an improved heuristic for solving the matrix bandwidth minimization problem based on genetic programming. The computational experiments show that our approach behaves very well in comparison with five of the best metaheuristics proposed for MBMP in terms of solution quality. We plan to improve the running times of our heuristic by developing a parallel implementation strategy of the algorithm.

Acknowledgments. This work was cofinanced from the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013, project number POSDRU/89/1.5/S/56287 "Postdoctoral research programs at the forefront of excellence in Information Society technologies and developing products and innovative processes", partner University of Oradea.

References

1. Corso, G.D., Manzini, G.: Finding exact solutions to the bandwidth minimization problem. Computing 62(3), 189–203 (1999)
2. Cuthill, E., McKee, J.: Reducing the bandwidth of sparse symmetric matrices. In: Proc. 24-th Nat. Conf., pp. 157–172. ACM, New York (1969)
3. Koohestani, B., Poli, R.: A Genetic Programming Approach to the Matrix Bandwidth-Minimization Problem. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 482–491. Springer, Heidelberg (2010)
4. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge (1992)
5. Lim, A., Rodriguez, B., Xiao, F.: Heuristics for matrix bandwidth reduction. European Journal of Operational Research 174(1), 69–91 (2006)
6. Lim, A., Lin, J., Xiao, F.: Particle swarm optimization and hill climbing for the bandwidth minimization problem. Applied Intelligence 26, 175–182 (2007)
7. Marti, R., Campos, V., Pinana, E.: A branch and bound algorithm for the matrix bandwidth minimization. European Journal of Operational Research 186, 513–528 (2008)
8. Marti, R., Laguna, M., Glover, F., Campos, V.: Reducing the bandwidth of a sparse matrix with tabu search. European Journal of Operational Research 135(2), 211–220 (2001)
9. Matei, O.: Evolutionary Computation: Principles and Practices. Risoprint (2008)
10. Mladenovic, N., Urosevic, D., Perez-Brito, D., García-González, C.G.: Variable neighbourhood search for bandwidth reduction. European Journal of Operational Research 200, 14–27 (2010)
11. Papadimitriou, C.H.: The NP-completeness of the bandwidth minimization problem. Computing 16, 263–270 (1976)
12. Pinana, E., Plana, I., Campos, V., Marti, R.: GRASP and path relinking for the matrix bandwidth minimization. European Journal of Operational Research 153, 200–210 (2004)
13. Rodriguez-Tello, E., Jin-Kao, H., Torres-Jimenez, J.: An improved simulated annealing algorithm for bandwidth minimization. European Journal of Operational Research 185, 1319–1335 (2008)

About Inducing Simple Emergent Behavior in Large Cournot Games by Using Crowding Based Differential Evolution

Rodica Ioana Lung

Babes-Bolyai University of Cluj Napoca, Romania,
Faculty of Economics and Business Administration
rodica.lung@econ.ubbcluj.ro

Abstract. In dealing with real-world large games it has been argued that some players may not be affected by the actions of all other players, i.e. that in a given situation of the game a player may be inclined to believe that only a subset of the rest of the players has a real influence on his payoff. In this paper some numerical experiments simulating such a situation using a Crowding based Differential Evolution algorithm adapted for detecting Nash equilibria are presented.

1 Introduction

In a continuous era of globalization one wonders how much ones actions and results are affected by agents acting at a relatively high distance. From a theoretical point of view, in large non-cooperative games - involving hundreds of players - each player's action is supposed to take into account the actions off all other players. But sometimes, due to geographical or other physical constraints, this is not possible. What happens in a large game when players decide to only take into account the actions of a neighboring set of players? What happens if, in different situations of a game, players decide to switch opponents according to personal criteria? What happens if their decision is overridden by a small group of major players that influence the entire game? While studying different types of interactions between players using a differential evolution algorithm, a micro-macro effect causing a simple type of emergent behavior is observed.

After a short introduction about the emergence phenomenon in Section 2 the generative relation used for computing Nash equilibria of noncooperative games is described in Section 3. Three types of interactions between players are proposed in Section 4. Numerical results presented in Section 7 are obtained using the Adapted Crowding Differential Evolution algorithm described in Section 6 for several instances of the Cournot oligopoly (Section 5).

2 Emergence

Emergence is a concept widely used in sciences, the arts and engineering. A short description would state that "emergence" is the notion that the whole is not the sum of its parts [6]. For example, flocks of birds flying in lockstep formation and schools of fish

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 75–83, 2011.

swimming in coherent array abruptly turn together with no leader guiding the group [3]. The emergence of order and organization in systems composed of many autonomous entities or agents is a very fundamental process.

Several attempts to classify and formalize the concept of emergence have been made [5]. In [7], [1] and [4] formalization attempts using grammars, formal languages and mathematics are proposed. The ubiquitousness of emergence in various, very different, fields prohibits - for the moment - a unified formalization that would fit every form of emergence encountered. However, one of the most important common characteristics across these fields that can be outlined [16] is represented by the micro-macro effect, which refers to the properties, behaviors, structures or patterns appearing at a higher macro level that cannot be explicitly found at the lower, micro level. Such an effect is induced and studied in this paper. Thus a micro-macro emergence effect observed in Cournot oligopolies where players are allowed to interact with each other - while keeping the non-cooperative nature of the game - by switching opponents is presented.

3 Generative Relations for Nash Equilibria

A generative relation for Nash equilibria is a relation between two strategy profiles that enables their comparison with respect to the Nash solution concept, i.e. it evaluates if one is "closer' to equilibrium. In [8] such a generative relation has been formally introduced.

3.1 Nash Ascendancy Relation

A finite strategic game is defined by $\Gamma = ((N, S_i, u_i), i = 1, n)$ where:

- N represents the set of players, $N = \{1, ..., n\}$, n is the number of players;
- for each player $i \in N$, S_i represents the set of actions available to him, $S_i = \{s_{i_1}, s_{i_2}, ..., s_{i_{m_i}}\}$ where m_i represents the number of strategies available to player i and $S = S_1 \times S_2 \times ... \times S_N$ is the set of all possible situations of the game;
- for each player $i \in N$, $u_i : S \to \mathbb{R}$ represents the payoff function.

Denote by (s_{i_j}, s^*_{-i}) the strategy profile obtained from s^* by replacing the strategy of player i with s_{i_j} i.e.

$$(s_{i_j}, s^*_{-i}) = (s^*_1, s^*_2, ..., s^*_{i-1}, s_{i_j}, s^*_{i+1}, ..., s^*_1).$$

The most common concept of solution for a non cooperative game is the concept of Nash equilibrium [10,11]. A collective strategy $s \in S$ for the game Γ represents a Nash equilibrium if no player has anything to gain by changing only his own strategy.

Consider two strategy profiles x and y from S. An operator $k : S \times S \to \mathbb{N}$ that associates the cardinality of the set

$$k(x, y) = |(\{i \in \{1, ..., n\} | u_i(y_i, x_{-i}) \geq u_i(x), y_i \neq x_i\}|$$

to the pair (x, y) is introduced.

This set is composed by the players i that would benefit if - given the strategy profile x - would change their strategy from x_i to y_i.

Let $x, y \in S$. We say the strategy profile x *Nash ascends* the strategy profile y in and we write $x \prec y$ if the inequality

$$k(x, y) < k(y, x)$$

holds.

Thus a strategy profile x ascends strategy profile y if there are less players that can increase their payoffs by switching their strategy from x_i to y_i than vice-versa. It can be said that strategy profile x is more stable (closer to equilibrium) then strategy y.

Two strategy profiles $x, y \in S$ may have the following relation:

1. either x dominates y, $x \prec y$ $(k(x, y) < k(y, x))$;
2. either y dominates x, $y \prec x$ $(k(x, y) > k(y, x))$;
3. or $k(x, y) = k(y, x)$ and x and y are considered indifferent (neither x dominates y nor y dominates x).

The strategy profile $s^* \in S$ is called non-ascended in Nash sense (NAS) if

$$\nexists s \in S, s \neq s^* \text{such that } s \prec s^*.$$

In [8] it is shown that all non-ascended strategies are NE and also all NE are non-ascended strategies. Thus the Nash ascendancy relation can be used to characterize the equilibria of a game.

4 Proposed Players Interactions

The following generalization of a non-cooperative game may be considered as $\Gamma = (N, S, U, R)$ where:

- N represents the set of players, $N = \{1, ..., n\}$, n is the number of players;
- for each player $i \in N$, S_i represents the set of actions available to him, $S_i = \{s_{i_1}, s_{i_2}, ..., s_{i_{m_i}}\}$ where m_i represents the number of strategies available to player i and $S = S_1 \times S_2 \times ... \times S_N$ is the set of all possible situations of the game;
- $R = \{r_1, ...r_d\}$ is the set of possible preferences for the players. For each situation s of the game we define $R_i(s) : S \to R$, $R_i(s) = r_k$ represents the preference of player i for all other players j having $R_j(s) = r_k$;
- for each player $i \in N$, $u_i : S \times R_i(S) \to \mathbb{R}$ represents the payoff function, $U = (u_1, ..., u_n)$;

Considering a situation $s \in S$ of the game, a player j having the connection $r_j(s)$ is considered connected to all players k having the same connection value $r_k(s) = r_j(s)$. Players connected to each other may erroneously consider that the game is limited only to their connections and compute their payoffs according to this belief. Thus its payoff function $u_j(s)$ will only take into account players that are connected to j. This approach represent a more realistic model of a large game as real players may not bother to take into account all their opponents or they may not be aware who all the players involved in the game are.

Three types of interactions among players are studied in this paper.

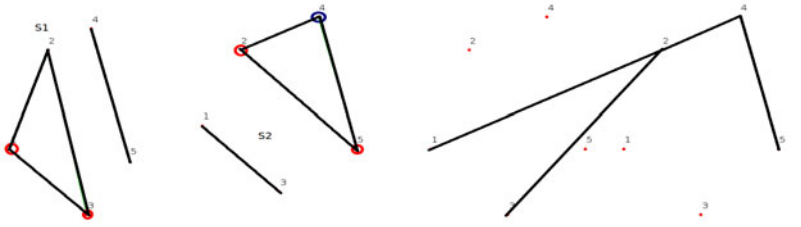


Fig. 1. Left: Two situations of the game with different connections between players. Player 4 in S2 is connected to players 2 and 5. In S1 player 2 is connected to player 1 and 3. These connections result in player 4 being influenced in some measure by players 2 and 5 but also indirectly, when comparing situation S2 to S1 to players 1 and 3 (right).

Type A. The simplest type of interaction is to consider that players are related to each other in a 'static' manner. The relationship between them is set beforehand and no changes are made, i.e. $r_i(s') = r_i(s''), \forall i \in \{1, ..., n\}$ and all $s', s'' \in S$.

Type B. Players connections differ from situation to situation. When faced with a choice, if allowed, a player will chose the connections that seem to 'promise' the highest payoff. Figure 1 illustrates interactions between five players in two situations of a game. When comparing two situations of a game the connections between players within both situations affect the entire game.

Type C. In real world games, although players may choose, either aware of the mistake or not, which are their 'worthy' opponents, there are some major players in the game that cannot be ignored and whose actions affect the payoffs of all other players.

Using the Nash ascendancy relation and Players Interactions. Although players may 'believe' to find themselves within a certain neighborhood, when comparing two situations of the game the 'real' evaluation is taken into account without considering any kind of relationship between players (which is equivalent to considering that they are all related in this case).

5 The Cournot Oligopoly

The Cournot oligopoly model was proposed by Antoine Augustine Cournot [2] and has since then been used as a reference model in economic applications [15].

Let q_i, $i = 1, ..., N$ denote the quantities of an homogeneous product - produced by N companies respectively. The market clearing price is

$$P(Q) = a - Q,$$

where Q is the aggregate quantity on the market. Hence we have

$$P(Q) = \begin{cases} a - Q, & \text{for } Q < a, \\ 0, & \text{for } Q \geq a. \end{cases}$$

Let us assume that the total cost for the company i of producing quantity q_i is $C(q_i) = cq_i$. Therefore, there are no fixed costs and the marginal cost c is constant, $c < a$. Suppose that the companies choose their quantities simultaneously. The payoff for the company i is its profit, which can be expressed as:

$$u_i(q_1, q_2, ..., q_N) = q_i P(Q) - C(q_i).$$

If we consider

$$Q = \sum_{i=1}^{N} q_i,$$

then

$$u_i(q_1, ..., q_N) = q_i(a - (q_1 + q_2 + ... + q_N) - c),$$

and there is one Nash equilibrium that can be computed by

$$q_i = \frac{a-c}{N+1}, \forall i \in \{1, ..., N\}.$$

Apart from its applications in economy the Cournot oligopoly model can be used to test the behavior of different evolutionary approaches computing Nash equilibria for a large number of players.

In our case, if Type A or Type B of connections are considered then we can write

$$u_i(q_1, ..., q_N) = q_i(a - (q_1\rho_1 + q_2\rho_2 + ... + q_N\rho_N) - c),$$

where

$$\rho_j = \begin{cases} 1, r_i(q_1, ..., q_N) = r_j(q_1, ..., q_N) \\ 0, r_i(q_1, ..., q_N) \neq r_j(q_1, ..., q_N) \end{cases}.$$

In the case of Type C of connections, if we denote by $r^* \in R$ the set of the major players of the game, we have:

$$u_i(q_1, ..., q_N) = q_i(a - (q_1\rho_1 + q_2\rho_2 + ... + q_N\rho_N) - c),$$

and

$$\rho_j = \begin{cases} 1, r_i(q_1, ..., q_N) = r_j(q_1, ..., q_N) \vee r_j(q_1, ..., q_N) = r^* \\ 0, r_i(q_1, ..., q_N) \neq r_j(q_1, ..., q_N) \end{cases}.$$

Thus major players consider that their payoffs are only affected by other major players while all other players are affected by the actions of any major player.

Because the Cournot oligopoly model is a symmetric one an asymmetric version is considered where the payoff function for player i is computed by

$$u_i(q_1, ..., q_N) = q_i(a + i - (q_1\rho_1 + q_2\rho_2 + ... + q_N\rho_N) - c).$$

6 Adapted Crowding Differential Evolution (ACDE)

Crowding Differential Evolution (CrDE) [14] which extends the Differential Evolution (DE) algorithm [12] with a crowding scheme has already been used for Nash equilibria detection [9].

Individuals evolved by ACDE represent situations of the game. Each situation of the game is composed by the strategies chosen by each player.

Regarding the crowding scheme, the only modification to the conventional DE is made regarding the individual (parent) being replaced. Usually, the parent producing the offspring is substituted, whereas in CrDE the offspring replaces the most similar individual among the population if it *Nash ascends* it. A *DE/rand/1/exp* [13] scheme is used.

The adapted version of the CrDE method takes into account possible player interactions when creating an offspring using the DE scheme. Thus the offspring will inherit the connection from the parent with the highest payoff.

7 Numerical Simulations

Numerical simulations are performed for Cournot games with 12, 30, 90 and 300 players. A maximum number of $3 \cdot 10^6$ payoff functions evaluations is set for a population of 50 players. The crowding factor is set to 50, $F = 0.01$ and $p_c = 0.9$. A player may choose among $d = 3$ connections, $R = \{0, 1, 2\}$ with an extra type of connection for Type C experiments.

7.1 Type A Interactions

For all situations of the game players interactions are constant throughout the evolutionary process. In all experiments interconnected players form sub-games evolving toward the Nash equilibrium of the sub-game. Table 1 presents average distance to Nash equilibria of the global game. Thus, if in evaluating the Nash ascendancy relation the entire set of players is taken into account, the ACDE converges to the NE of the game, but significantly slower. Results concerning convergence to the Nash equilibrium in large Cournot games can be found in [9].

Table 1. Type A interactions. Average distance to Nash equilibria of the $d = 3$ sub-games formed.

	Cournot		Modified Cournot	
No of players	Avg Dist to NE	St Dev	Avg Dist to NE	St Dev
12	1.25	0.38	1.73	0.52
30	6.45	0.55	7.45	0.60
90	10.63	1.29	14.03	1.01
300	47.29	1.54	48.16	1.67

Table 2. Average number of players grouped in sub-games for the Cournot oligopoly and the modified asymmetric Cournot oligopoly

	Cournot		Modified Cournot	
No of players	Avg No. players in a subgame	St Dev	Avg No. players in a subgame	St Dev
12	4	0	4	0
30	10	0.2	10	0.4
90	30	2.3	30	2.41
300	100	3.58	100	3.87

7.2 Type B Interactions

For each player in each individual in ACDE a random number between 0 and $d-1$ is generated. All players having the same number consider to be connected to each other and their payoff is computed considering only players connected to each other. However, different individuals will have different sets of connections. This can be interpreted by the fact that players may change their beliefs regarding their opponents in different situations of the game. In this case the ACDE population may be regarded as a complex network. At a micro-level, within each individual players evolve their connections, while at a macro level sub-games with *equal* number of players emerge. This emergent behavior is illustrated by the fact that in all experiments number of players connected to each other converges towards N/d, i.e the total number of players divided by the number of connection possibilities among players. Table 2 presents the average number of players belonging to the same sub-game with the standard deviation of that number.

7.3 Type C Interactions

When dealing with major players two approaches have been considered. The first one allows players to become major players during the offspring creation process if one of the major players is selected as a parent and if its payoff is higher than the payoff of the other two parents selected. In this case, in all runs and for all considered number of players, all players are converted to major players, which would be expected and may be explained by the fact that by the definition of their payoff functions their payoffs will be higher than that of the other players. However, if the set of the major players is, in a more realistic approach, considered to be an exclusive one and no other players may enter it, then the same kind of emergent behavior as observed in the case of type B interactions appears - the other players form subgames of equal sizes as illustrated in Table 3. The number of major players considered is 10% of the total number of players.

Table 3. Number of players in sub-games formed by type C interactions

		Cournot		Modified Cournot	
No of players	No. of major pl.	Avg. no. players/sub-game	St Dev	Avg. no. players/sub-game	St Dev
12	3	3	0	3	0
30	3	9	1.3	9	1.5
90	9	27	2.13	27	2.87
300	30	90	3.59	90	3.81

8 Conclusions

A simple type of emergent behavior in large games obtained by considering different types of interactions among players involved in large Cournot oligopolies are presented. When players are allowed to interact with each other (but not cooperate) sub-games of equal size are formed. This result indicates that the equilibrium state when considering relations among players is achieved when players group themselves in similar games. Although for type B interactions sub-games are formed, these are interconnected games and their equilibrium is analytically difficult to compute. Further work consists in studying different kinds of interactions and possible emergent behavior in large games and using them in real economic settings.

Acknowledgments

This research is supported by Grant TE 320 - Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCSIS, Romania.

References

1. Bar-Yam, Y.: A mathematical theory of strong emergence using multiscale variety. Complex. 9, 15–24 (July 2004), http://portal.acm.org/citation.cfm?id=1041069.1041073
2. Cournot, A.A.: Recherches sur les principes mathematiques de la theorie des richesses [microform] / par Augustin Cournot. L. Hachette, Paris (1838), http://galenet.galegroup.com/servlet/MOME?locID=nla
3. Crutchfield, J.P.: The calculi of emergence: Computation, dynamics, and induction. Physica D 75, 11–54 (1994)
4. Deguet, J., Magnin, L., Demazeau, Y.: Elements about the emergence issue: A survey of emergence definitions. ComPlexUs 3, 24–31 (2006)
5. Fromm, J.: Types and forms of emergence (2005)
6. Holland, J.: Emergence: From Chaos to Order. Oxford University Press, Oxford (1998)
7. Kubík, A.: Toward a formalization of emergence. Artif. Life 9, 41–65 (December 2002), http://portal.acm.org/citation.cfm?id=778855.778861
8. Lung, R.I., Dumitrescu, D.: Computing nash equilibria by means of evolutionary computation. Int. J. of Computers, Communications & Control III (suppl.issue), 364–368 (2008)
9. Lung, R.I., Mihoc, T.D., Dumitrescu, D.: Nash equilibria detection for multi-player games. In: IEEE Congress on Evolutionary Computation, pp. 1–5 (2010)
10. McKelvey, R.D., McLennan, A.: Computation of equilibria in finite games. In: Amman, H.M., Kendrick, D.A., Rust, J. (eds.) Handbook of Computational Economics, vol. 1, ch. 2, pp. 87–142. Elsevier, Amsterdam (1996)
11. Nash, J.F.: Non-cooperative games. Annals of Mathematics 54, 286–295 (1951)
12. Storn, R., Price, K.: Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Tech. Rep. TR-95-012, Berkeley, CA (1995), http://citeseer.ist.psu.edu/article/storn95differential.html
13. Storn, R., Price, K.: Differential evolution a simple evolution strategy for fast optimization. Dr. Dobb's Journal of Software Tools 22(4), 18–24 (1997)

14. Thomsen, R.: Multimodal optimization using crowding-based differential evolution. In: Proceedings of the 2004 IEEE Congress on Evolutionary Computation, Portland, Oregon, June 20-23, pp. 1382–1389. IEEE Press, Portland (2004)
15. Tirole, J.: The Theory of Industrial Organization, MIT Press Books, vol. 1. The MIT Press (June 1988), http://ideas.repec.org/b/mtp/titles/0262200716.html
16. Wolf, T.D., Holvoet, T.: Emergence versus self-organisation: Different concepts but promising when combined, pp. 1–15. Springer, Heidelberg (2005)

An Study of the Tree Generation Algorithms in Equation Based Model Learning with Low Quality Data

Alba Berzosa[1], José R. Villar[2], Javier Sedano[1], Marco García-Tamargo[2], and Enrique de la Cal[2]

[1] Instituto Tecnológico de Castilla y León, Lopez Bravo 70, Pol. Ind. Villalonquéjar 09001 Burgos, Spain
{alba.berzosa,javier.sedano}@itcl.es
[2] Computer Science Department, University of Oviedo, Campus de Viesques s/n 33204 Gijón, Spain
{villarjose,marco,delacal}@uniovi.es

Abstract. The undesired effects of data gathered from real world can be produced by the noise in the process, the bias of the sensors and the presence of hysteresis, among other uncertainty sources. In previous works the learning models using the so-called Low Quality Data (LQD) has been studied in order to analyze the way to represent the uncertainty. It makes use of genetic programming and the multiobjective simmulated annealing heuristic, which has been hybridized with genetic operators. The role of the tree generation methods when learning LQD was studied in that paper. The present work deals with the analysis of the generation methods relevance in depth and provides with statistical studies on the obtained results.

Keywords: Genetic Programming, Genetic Algorithm and Programming, Low Quality Data, Multiobjective Simulated Annealing.

1 Introduction

With the scarce energy sources and the worsening environmental pollution, how to use the existing energy is becoming a very important challenge in various fields of modern engineering [8,6,16]. For example, notorious efforts have been made within the area of lighting control systems, whose aim is to control the electrical power consumption for the ballast in the installation so the luminance complies with the regulations. In [13,15] a lighting control system was considered to show the relevance of the uncertainty for an efficient energy use. The typical control loop includes a light sensor, the light ballasts and a light controller. The sensors measure the amount of light in a room, but they have some drawbacks: they operate with hysteresis and saturation [6] and their measurements depend on the light sensor unit. In the studied literature, when obtaining models for simulation, only crisp values are regarded as the measurements from light sensors. Obviously, the inputs and outputs of the light sensor models obtained

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 84–91, 2011.

are also crisp variables. But several studies have presented the decrease in the performance of crisp algorithms as data uncertainty increases [5].

An approach for learning white box models when LQD is available is presented in [15,2], where the variables are assumed with an unknown uncertainty value modelled as a fuzzy number. The white box models include an equation -represented as a nodes tree- setting the relationship of the output variable with the feature space and a set of constants. Some of the constants are use to model the vagueness of the variables and others are used as terminal nodes in the equation. A genetic programming hybridized with Genetic Algorithm (hereafter GAP) which is evolved by means of a Multi Objective Simulated Annealing algorithm (hereinafter MOSA) is proposed, and a random tree generation algorithm to create the individuals is carried out in the evolutionary algorithm. A MOSA hybridized with genetic operators is proposed (hereinafter, SAP), and a random tree generation algorithm to create the individuals is carried out in the evolutionary algorithm. The results show that the proposed algorithm remails with the same performance index even though LQD is given. The relevance of three generation algorithms in MOSA hybridized with genetic operators is studied in [2]. The approach makes use of fuzzy fitness funcions. Consequently, the relevance of the tree generation methods is focused on the first generations due to the temperature grading and the need to search in the whole variables space. Two algorithms were compared. On the one hand, the so-called GROW GP tree generation algorithm, described in [7], chooses a node type with equal probability, including the terminal and non-terminal ones. On the other hand, [9] offers an alternative tree-creation algorithm, the Probabilistic Tree Creation 1 (PTC1), which gives the user control over the expected tree size, the maximum deth and the probabilities of appearance of functions within the tree, providing in addition, very low computational complexity.

The present work aims to extend it with the representation and comparison of the model learning evolution when genetic programming and uncertainty are considered for the two tree generation methods proposed, GROW and PTC1. For this purpose, an statistical study of the results has been done. The remainder of this manuscript is as follows. Firstly, a description of the simulated annealing approach for learning white box models with LQD is included. Then, the different tree generation methods employed in this comparison are detailed. The experiments and the results are discussed next, while in the last section the conclusions are drawn and the future work is proposed.

2 White Box Models SAP Learning with LQD

Soft Computing includes the set of techniques that allow learning models using the knowledge in the data [1,3,11,17]. There are several uncertainty sources in data gathered from real processes [4]. In this work, we study data which has been gathered as crisp data but that are highly imprecise, i.e., the data gathered from light sensors [6,13]. In [15] a SAP approach for learning white box models from this kind of data is presented.In that study, the representation of the

vagueness in a GP model is represented by two constants C^- and C^+ assigned to each imprecise variable which evolves in the learning process. These constants represent the limits of a triangular membership function for an α-cut$= 0$ which is associated with each imprecise variable. Let us suppose the training data set being the set $\{d_i^j\}$, with $i = \{0, \ldots, D\}$ for each of the D variables X_i and $j = \{1, \ldots, N\}$ and N the number of examples in the data set. Then, whenever an imprecise variable X_i is evaluated for the example j, a fuzzy number with a triangular fuzzy membership defined through the three following values $[d_i^j - C^-, d_i^j, d_i^j + C^+]$ is returned. If symmetrical membership functions are considered, only one constant per imprecise variable is needed. As in classical fuzzy logic literature, crisp values from constants or from crisp variables are extended to fuzzy singletons, so only operations with fuzzy numbers are required. In order to reduce the computational cost, the solution presented in [14] is used, and evaluations are calculated only for certain predefined α-cuts.

An individual in this study is a compound of the equation representation, the constants vector and the specification of the uncertainty, which is provided with the number of constants used to represent the uncertainty and the list of indexes of the input variables in the dataset that are supposed to manage LQD. As in Genetic Programming hybridized with Genetic Algorithms (hereinafter, GAP) models, the equation representation consists of a nodes tree, each internal node corresponds with a valid operator, and the leaf nodes correspond with a variable or a constant index. The number of constants is predefined, so the constant vector in all individuals has the same size. The first group of constants in the constant vector is assigned to the uncertainty management. The generation of a nodes tree is the well-known random strategy given by the GROW method [7].

Evolving GAP individuals introduce four genetic operators, two come from GP evolution (the GP crossover and mutation) and two come from GA (GA crossover and mutation). The GP operators introduce variability in the structure of the model, in other words, the equation itself. The GA operators modify the vector of constants. In all the cases, there is a predefined probability of carrying out each of these genetic operators. In each run, the type of operation to carry out is chosen, that is only GP or GA operations can take place, but never both in the same run. The fitness of an individual is calculated as the mean squared error, which in fact is a fuzzy number. To reduce the width of the intervals and to obtain models that include the desired output crisp data two more objectives have also been considered, so multi-objective techniques are needed. The evolutionary algorithm is the MOSA proposed in [10,14].

3 Tree-Generation Algorithms

Tree-creation plays an important role in GP algorithms since a good random tree-creation is needed to create the number of trees that will compose the initial population and the subtrees used in subtree mutation. Besides, as stated in [9], tree creation is also related with tree bloat, that is, the tendency of GP trees to grow during the evolutionary process [12] which causes the slowdown of the evolutionary process by making individuals more resistant to change. This

section discusses the role of the tree generation methods when learning LQD. The above mentioned SAP approach is used for learning models with such kind of data, and two different techniques for tree generation are compared: the GROW and the PTC1 methods.

In [7] the so-called GROW GP tree-generation algorithm has been widely used since its formulation despite having several weaknesses. Although originally not designed to control the depth and the size of the tree, it can be easily extended to do so. To generate a tree, the algorithm chooses a node type with equal probability. The choice of a node includes the terminal and non-terminal nodes. Once a node has been chosen, and attending to the arity of the node, the algorithm moves to each of its operands and is executed recursively. This process continues either until all the leaf nodes are terminal nodes, or either the depth limit or the tree size limit is reached. In these two latter cases, the tree is filled with terminal nodes.

On the other hand, [9] propose an alternative tree-creation algorithm, the Probabilistic Tree Creation 1 (PTC1). This algorithm gives the user control over the expected tree size, a method parameter. Instead of attempting to generate completely uniformly distributed tree structures, PTC1 allows user-defined probabilities of appearance of functions within the tree plus a desired maximum depth D, providing in addition, very low computational complexity. However, PTC1 does not provide any control over the variance in tree size generated, which limits its usefulness. In this algorithm, the set of functions F is divided into two disjoint subsets: terminals nodes set T, each one with probability q_t, and non-terminals nodes N, each one with probability q_t. The recursive method chooses between terminal and non-terminal nodes type for the current node with probability p (see Eq. 1), where b_n is the arity of non-terminal n. When a terminal node type is chosen, variable or constant is decided according to q_t. In case of non-terminal nodes, the node type will be selected according to q_n. Each of its operators is chosen recursively. Both $\{q_t\}$ and $\{q_n\}$ are given by the user.

$$p = \frac{1 - \frac{1}{E_{tree}}}{\sum_{n \in N} q_n b_n} \tag{1}$$

4 Experimentation and Results

In the experimentation stage, the performance of GROW and PTC1 algorithms when learning models with LQD is compared when both regression and time series problems are faced. Several different synthetic LQD data set are generated. In order to obtain such LQD, a two step procedure has been carried out: firstly, the crisp data sets have been generated and then the uncertainty have been introduced to the data. Four different problems are proposed, three of them correspond with regression problems and one is a time series problems. In regression problems the model can not be a function of the output variable, while in the time series problems this variable can be included in the equations. The

output variable should be included in an equation and it must be affected with a delay operator at least. Time series problems models should be evaluated recursively, that is, the output at time t should be calculated using the previously calculated values of the output variable. The regression problems are defined through f_1, f_2 and f_3 in Table 1, while f_4 represents a time series problem. In all the cases, four input variables are considered ($\{x_0, x_1, x_2, x_3\}$. The variable time t is used to calculate the values of the examples $\{x_i, \forall i\}$. It is included the output variable calculated with the corresponding formulae f_i. The second step for generating the LQD data sets is the aggregation of uncertainty to the data. For each variable in an example, a random value in the range $[-1e^{-4}, 2e^{-4}]$ is added. All the involved variables in a data set are affected, including the output variable, so that all the variables are parameterized as imprecise. The parameters used are presented in Table 2. Although the number of iterations is set to 1000, data has been gathered each 200 iterations in order to determine the fitness evolution during the simulated annealing run.

Table 1. Formulae for the data sets generation

$f_1 = x_1 + x_0 * (x_2 - 0.5)$	$t = \{1, \ldots, 100\}$
$f_2 = 2 * x_1 * x_2$	$x_0 = abs(\cos(t))$
$f_3 = \cos(x_0) * (x_2 - x_1)$	$x_1 = abs(\sin(t))$
$f_4 = 2 * x_2 * delay(f_4)$	$x_2 = abs(\cos(t) * \sin(t))$
	$x_3 =$ random in the range $[0, 1]$

Table 2. Parameters used in the experimentation stage. When LQD is assumed, all the variables are set as imprecise variables. In all the cases symmetrical triangular membership functions are used, so only one constant in the GA constants vector is needed per variable. Whenever time series learning is carried out the models are evaluated recursively.

Parameter	Value	Parameter	Value
$\alpha-$cut	0.95	population size	50
Constants range	[-10,10]	GP mutation prob.	0.25
$C^- == C^+$ range	[0, 0.01] %	GP crossover prob.	1
SA Δ value	0.1	GA mutation prob.	0.5
SA initial temperature	1	GP crossover prob.	1
SA final temperature	0	stop generation	{50, 100, 300}
PTC1 Constant prob.	0.3	generations	1000
PTC1 Operators probability	+ 0.145 - 0.145 * 0.145 / 0.145 max 0.08 min 0.08 delay 0.1 sin 0.08 cos 0.08		
N^o of GA constants	7	Maximum size	5
Maximum depth	5		

For both the GROW and the PTC1 methods with each of the four datasets, ten runs have been carried out. Results are shown in Table 3 and Figure 1. The

Table 3. Mean of the MSE values for the individuals with the best value of the fitness sum in each of the ten runs performed for the four datasets. Results shown should be multiplied by 10^{-3}.

	200		400		600		800		1000	
Dataset	GROW	PTC1	GROW	PTC1	GROW	PTC1	GROW	PTC1	GROW	PTC1
f_1	1.153	1.197	1.208	1.336	1.166	1.391	1.166	1.480	3.613	1.480
f_2	12.254	3.096	11.677	2.751	11.637	2.751	11.557	3.076	0.628	1.611
f_3	0.038	0.190	0.038	0.179	0.042	0.299	0.042	0.299	0.041	0.629
f_4	2.529	2.591	3.076	2.440	2.359	2.206	2.170	2.490	2.138	2.553

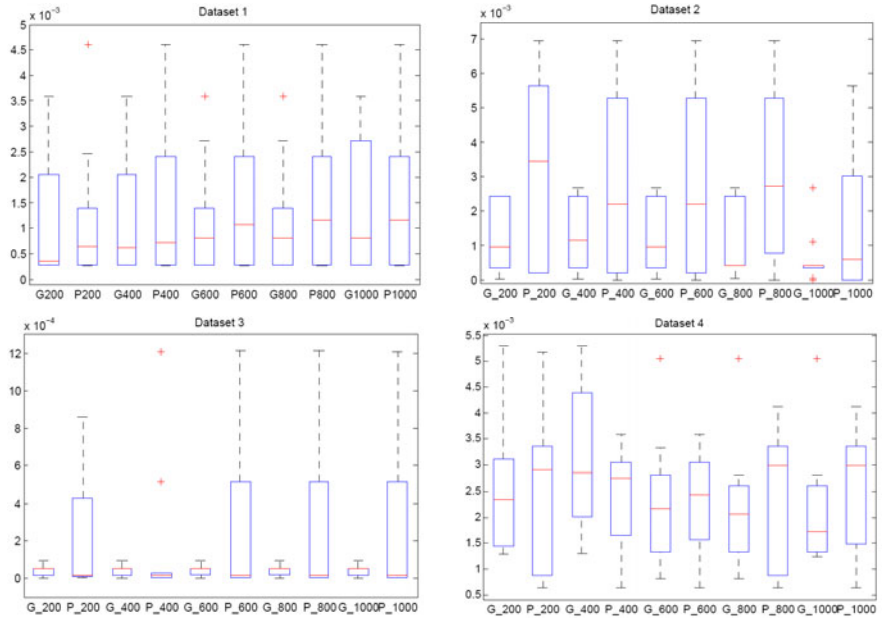


Fig. 1. Boxplots for the values of the individuals with the best value of the fitness sum in each of the ten runs performed for the four datasets

former shows the values for the means of the MSE of the individuals with the best value for the sum of squared values for the considered fitness in the MOSA: the MSE (mean squared error), number of covered examples and mean output width. On the other hand, in the depicted boxplots, it is shown that PTC1 has a faster convergence than GROW as results are more stable during the succesive iterations. Values for the medians when GROW is used are slightly better than the ones for the PTC1 method, consequently, better results could be obtained if the number of iterations is increased.

5 Conclusions and Future Work

An statistical study on the results gathered from the run of two tree generation algorithms in SAP learning problems whith LQD has been studied. Data has been gathered each 200 iterations in order to study the fitness evolution. This study reveals that the choice of the suitable tree generation algorithm is relevant to the obtained results. The algorithms used are the GROW and the PTC1. Both behave in a similar way, nevertheless the PTC1 is shown to have a better convergence, although the medians values obtained when GROW is used are slighty better, so it can be assumed that if the number of iterations were increased, the performance would improve. Future work includes the study of the evolution of the diversity that each of the tree generation algorithms induces through the temperature evolution in the MOSA learning process, should be considered.

Acknowledgments. This research has been funded by the Spanish Ministry of Science and Innovation, under project TIN2008-06681-C06-04, the Spanish Ministry of Science and Innovation [PID 560300-2009-11], the Junta de Castilla y Len [CCTT/10/BU/0002] and by the ITCL project CONSOCO.

References

1. Alcalá-fdez, J., Sánchez, L., García, S., Del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F.: Keel: A software tool to assess evolutionary algorithms for data mining problems?
2. Berzosa, B., Villar, J.R., Sedano, M., García-Tamargo, J.: Tree generation methods comparison in gap problems with low quality data. In: Accepted for publication in the Proceedings of the International Conference on Soft Computing Models in Industrial and Environmental Applications SOCO 2011. AISC (2011)
3. Corchado, E., Herrero, A.: Neural visualization of network traffic data for intrusion detection. Applied Soft Computing (2010)
4. Ferson, S., Kreinovich, V., Hajagos, J., Oberkampf, W., Ginzburg, L.: Experimental uncertainty estimation and statistics for data having interval uncertainty. Technical report, Technical Report SAND2007-0939 (2007), http://www.ramas.com/intstats.pdf
5. Folleco, A., Khoshgoftaar, T.M., Hulse, J.V., Napolitano, A.: Identifying learners robust to low quality data. Informatica (Slovenia) 33(3), 245–259 (2009)
6. De Keyser, R., Ionescu, C.: Modelling and simulation of a lighting control system. Simulation Modelling Practice and Theory 18(2), 165–176 (2010)
7. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
8. Li, D.H.W., Cheung, K.L., Wong, S.L., Lam, T.N.T.: An analysis of energy-efficient light fittings and lighting controls. Applied Energy 87(2), 558–567 (2010)
9. Luke, S.: Two fast tree-creation algorithms for genetic programming. IEEE Transactions on Evolutionary Computation 4(3), 274–283 (2000)
10. Sánchez, L., Rosario Suárez, M., Villar, J.R., Couso, I.: Mutual information-based feature selection and partition design in fuzzy rule-based classifiers from vague data. Int. J. Approx. Reasoning 49, 607–622 (2008)

11. Sedano, J., Curiel, L., Corchado, E., de la Cal, E., Villar, J.R.: A Soft Computing Method for Detecting Lifetime Building Thermal Insulation Failures. Integr. Comput. Aided Eng. 17, 103–115 (2010)
12. Soule, T., Foster, J.A., Dickinson, J.: Code growth in genetic programming. In: Proceedings of the First Annual Conference on Genetic Programming GECCO 1996, pp. 215–223. MIT Press, Cambridge (1996)
13. Villar, J.R., de la Cal, E., Sedano, J., García-Tamargo, M.: Analysing the low quality of the data in lighting control systems. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 421–428. Springer, Heidelberg (2010)
14. Villar, J., Otero, A., Otero, J., Sánchez, L.: Taximeter verification with gps and soft computing techniques. Soft Comput. 14, 405–418 (2009)
15. Villar, J.R., Berzosa, A., de la Cal, E., Sedano, J., García-Tamargo, M.: Multiobjecve simulated annealing in genetic algorithm and programming learning with low quality data. Submitted to Neural Computing (2010)
16. Villar, J.R., de la Cal, E., Sedano, J.: A fuzzy logic based efficient energy saving approach for domestic heating systems. Integrated Computer-Aided Engineering 16, 151–163 (2009)
17. Yu, W.-D., Sedano, L.Y.-C.: Ahybridization of cbr and numeric soft computing techniques for mining of scarce construction databases. Autom. in Constr. 15, 33–46 (2006)

RT-MLR: A Hybrid Framework for Context-Aware Systems

Pablo Rangel, José G. de Carvalho Jr., Milton R. Ramirez, and Jano M. de Souza

Brazilian Navy Research Institute
Rio de Janeiro, Brazil
{pablo,gomes}@ipqm.mar.mil.br
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
{pablorangel,milton,jano}@cos.ufrj.br

Abstract. This work presents a hybrid framework solution to describe context and develop context-aware systems. The framework intends to provide best domain knowledge expressiveness combining First Order Logic, Temporal Logic, and Fuzzy Logic. The framework engine was developed to be used in Object-Oriented systems, integrating the domain rules and system objects using an event-based architecture. The framework has been exemplified based on its capability to perform information fusion based on a heterogeneous context definition.

Keywords: Information Fusion, Context Representation, Context Reasoning.

1 Introduction

Context-aware systems have been developed since the 1990 decade [1]. A system is context-aware if can extract, interpret, use context information, and adapt its functionality to the current context of use [2]. The architecture of these systems needs to provide mechanisms to deal with large information and knowledge volumes, supported by sensors and users. Performing information fusion according to domain knowledge needs to be supported by a best context representation. For authors Baader et al. [3], the integration of heterogeneous information into a unique semantic level is a challenge that remains opened.

Besides the context representation issue, there is a concern about adaptability mechanisms. In many occasions, the essence of context-aware systems is related to monitoring and inference capability. The context reasoning mechanism for information fusion is critical for many reasons, such as user comfort, precision of results, and real-time requirements.

There are many ways for context representation and the development of context-aware systems [4]. The Object-Oriented (OO) approach and Rule-Based (RB) approach consist of a set of the most used techniques to represent knowledge and the development of context-aware systems into a complementary way.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 92–99, 2011.

Based on this scenario, this works presents a framework called Real-Time Multiple Logic Reasoner (RT-MLR), that aims to provide a solution for context representation and context reasoning, combining OO e RB approaches.

This work assumes that the expression of the rules using software objects and heterogeneous logics may improve context representation and context reasoning. With the RT-MLR solution, the rules can be written through three logic types: First-Order Logic, Temporal Logic, and Fuzzy Logic. Also, context monitoring through an event-based approach allows for adequate computational cost. The RT-MLR takes the advantage of the software objects and domain rules association, aiming to investigate only the rules associated with facts that have been changed in the context.

The RT-MLR has been exemplified based on its capability to perform information fusion of military tracks [5]. These applications are good examples of context-aware systems and real-time context reasoning.

This paper has four sections apart from the introduction itself. Section 2 discusses issues in the development of context-aware systems, and context representation. Section 3 presents the implemented framework. The case studies of a military context application and its results are shown in Section 4. Section 5 presents the conclusion and future work paths. Any comparison between the rule-based approach and other approaches (e.g. statistical approach) to perform data fusion is beyond this work.

2 Context Representation and the System Development Issue

Rules in context-aware systems have been used with different purposes since 1993 [6-7] until present days [8]. Unfortunately, the integration of RB and OO has some issues.

In the OO approach, the current state of the objects and relations represents the current context. Simple context-aware systems may be developed using the OO native resources [9]. The main idea in using OO to develop context-aware systems is the benefits of encapsulation and reutilization [4]. According to Bouzy and Cazenave [9], the strength in the OO approach lies in the simplification of knowledge representation of complex domains. Nevertheless, Java and other OO languages usually have no native resources to support a complete inference mechanism. It is possible to represent First Order Logic rules with Java operators. However, sometimes is necessary to aggregate more semantic value to the knowledge representation. Fuzzy Logic, for example, enables to express the knowledge of human experts in a more concise way. Temporal Logic, on the other hand, is useful for behaviours that can only be identified if one considers the events related to the time they occurred.

On the other hand, the RB approach consists of a widely accepted way to describe the domain knowledge. Context-aware systems are developed to act according to domain rules, which have a logical and formal property. In context-aware systems, the need for performing reasoning is aligned with the RB approach [10]. The main feature of RB approach is the high level of formality provided [4].

2.1 Integrating Software Objects with Formal Rules

A RB system usually has a database for the rules and facts. Whenever a rule or fact needs to be considered in a context investigation, it will be necessary to insert or

remove the rule or fact of its database. In contrast, when the rules are written with the software objects, the management of the rules is unavailable in execution-time. The rules are compiled with the source code of the system. The uses of software object to express the rules introduces the problems of code maintenance. Whenever a rules need to be considered in investigation, it will be necessary to re-compile the source code. Consequently, this program will probably present a best performance, because it will not be necessary to perform a search into a database.

The concept integration occurs less traumatic for facts database. There is nothing that does not allow the expression of rules through software objects. The only restriction can be the real-time requirements, because the actions of inserting and removing in a high change context may result in a high computational effort.

2.2 Context Monitoring

It is very important to deal with contexts that have system objects with constant changes. Continuous context monitoring is in line with real-time requirements. The reason for this is that rules need to be investigated whenever changes occur in the domain context. It is not enough to have resources to make inferences – it is also necessary to perform an automatic on-line monitoring of system objects, without direct instructions from the software programmer.

Continuous monitoring needs to cover all the situations the inference engine must be used in. Basically, there are two ways to make continuous monitoring [11]: through the pooling process, or based on specific events (event-based approach). The first one has an inherent problem that refers to computational performance, as the process consists of a cyclical evaluation of system object data. The second one has better computational performance, as the process consists of performing context reasoning according to determined related event (e.g. data modification).

3 The Framework

The Real Time Multiple Logic Reasoner (RT-MLR) has been designed to support context representation and context reasoning, combining the RB method with the regular OO architectures in a hybrid solution. A domain expert can express the knowledge trough multiple logic rules using the programming language syntax itself.

3.1 Assumptions

RT-MLR assumes an event-based approach, which means that rules are fired when significant changes in the domain occur. This is the key for the continuous monitoring capability of the RT-MLR. Each rule is described with system objects, naturally associating objects with rules. When some object changes its value, only associated rules are fired. Additionally, if a software engineer knows which rules can be applied for each event at design-time, fewer rules will be investigated. The OO approach is useful for both parts of system, application domain and RT-MLR. The OO approach also has an inherent reusability capability, which allows the development of applications by reusing previously written rules, just by using an association class.

In the RT-MLR the precedence is only set by the sequence in which the rules are added to a set of rules. Rules that are inserted later have higher precedence. As in the work of Dörflinger [12], the hierarchy solves the problem of mutual exclusion. In fact, for the First-Order Logic, the set exists only to define the precedence of the rules. However, in the case of fuzzy rules, each group should aggregate the rules that relate to the same output variables (variables that appear at the rule conclusion).

For First-Order Logic, the RT-MLR works with the closed world assumption in a non-monotonic logic and also with the unique name assumption. For Temporal Logic, the RT-MLR uses the concept of Interval Temporal Logic [13]. Applying Temporal Logic operators onto system objects means that a list of values with timestamp will be created for each associated object. Thus, the application does not need to keep values in runtime, as values are automatically memorized by the RT-MLR. Besides, Temporal Logic rules will be investigated as First-Order Logic, but the evaluation also will look for values and their timestamps in the RT-MLR memory. Finally, for Fuzzy Logic, rules are investigated like any other. The left side of each fuzzy rule is evaluated based on the domain objects and fuzzy operators. If the membership of the premise is greater than zero, the rule is fired, generating an output function for each fired rule. The output functions of the fired rules are aggregated and the final function is then deffuzyfied to produce a scalar value for the output variable.

3.2 Implementation

The software engineer can develop an application in the manner that is the most convenient. Using framework, the engineer has to design the classes that will be used in some rule, extending a class called Observable Data (OD). OD encapsulates Java basic types and provides a mechanism to automatically associate a rule to the object. The getters and setters methods of OD allow data changes concurrently. Additionally, it is possible to establish a confidence interval for attribute values to define the minimum variation accepted as relevant to rule investigation. Thus, small variances in value that are not considered relevant and associated rules will not be investigated.

Basically, a class Rule has two parts: premise and conclusion. Conclusion is an abstract class of framework, forcing the software engineer to extend Conclusion class and implementing a conclusion method. When premise is valid, conclusion is valid and method conclusion is called in. It is useful when one needs to program instructions according to the conclusion result.

Developing premise is quite simple, as premise is a combination of implemented operators (see table I). Each operator is a class of framework and it is only necessary to instantiate the operator object with another operator object, OD or Java instruction (fuzzy operators accept a special object argument called Fuzzy Set). Each rule needs to be part of one or more RuleSet object. There is nothing to do for the rule set of First-Order Logic and Temporal Logic rules. However, for Fuzzy Logic, a domain class needs to be created extending the class FuzzyRuleSet (FRS). FRS is abstract, which forces the software engineer to implement a fuzzyLogicRulesSetConclusion method. This method is called at the end of the deffuzyfication process.

It is also possible to combine different logic operators. The type of RuleSet will determine the nature of rules. For example, it is possible to write a rule combining Fuzzy Logic operators with First-Order Logic operators. However, the combination of

operators does not change the nature of the rule. Thus, even the First-Order Logic rule uses of a Fuzzy Logic operator, the nature of a rule will not be changed. In this case, for example, a deffuzyfication process will not be carried out.

Table 1. RT-MLR Operators

Logic Type	Operators Classes [Meaning]
First Order	Eq [=], Not [¬], Ht [>], He [≥], Lt [<], Le [≤], And [∧], Or [∨], XOr [⊕], Exists [∃(φ, L)], All [∀ (φ, L)].
Temporal	ThereWas [∃ (φ,T1, T2)], Count [∑ ∃ (φ , T1, T2)], Average [Avg(φ , T1, T2)], Occurs [∃ (φ1, before or after, φ2, T1, T2)].
Fuzzy	IsF [μS(X)], AndF [Min(μ(X), μ(Y))], OrF [Max(μ(X), μ(Y))], NotF[1 - μ(X)].

4 Case Studies

In military applications, decisions comply with doctrines that specify procedures in different situations. These procedures are based on data from different sources (sonar, radar, GPS etc.) and knowledge from experts. Data combination and fusion is usually performed to provide a better scenario understanding. In naval systems, tracks are representations of a detection of some entity (ships, aircrafts, helicopters, and so on). A track representation consists of estimated position, speed and course of radar dots.

Aiming to support the case studies, a simple simulation environment of a military application has been created with the RT-MLR. It is possible to simulate radar tracks that represent measurements made by two different radar nodes (R1 and R2). Track is a class in this environment and all attributes of it is an ObservableData framework class.

4.1 Correlation of Military Tracks

A well-known issue in data fusion in military contexts is to provide a correlation with tracks, as different radars often provide tracks that refer to the same entity. Military applications need to perform context reasoning if they wish to know whether tracks from different sources refer to the same entity (correlation).

Track to track correlation is made with rules that compare kinematic data of tracks. Rules are fired by reception of new data, but, previously to comparison, kinematic data of tracks are aligned in time. The following attributes are used: position (P), speed (S) and course (C). The correlation process is done by comparing the module of the difference (Δ) between each attribute of two tracks (T1 e T2) with a default value, in order to decide if the attributes from the different tracks belong to the same target.

$$|P_{T1} - P_{T2}| \leq 500 \text{ yards AND } |S_{T1} - S_{T2}| \leq 5 \text{ knots AND } |C_{T1} - C_{T2}| \leq 2\text{degrees} \rightarrow T1 = T2 \quad (1)$$

Three scenarios were created for context reasoning on the correlation problem. The first scenario has two vessels 5 miles apart and moving in intersection courses. The second scenario has a small vessel and a helicopter, in the same course. The third scenario has two ships in manoeuvring trajectories (fig. 1).

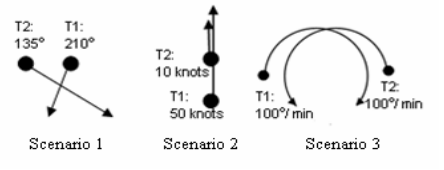


Fig. 1. Tested scenarios for track association (case study 1)

The tests were performed using two simulated radars. Each radar has two radar dots, generating four tracks (2 tracks per radar node). The rule tries to associate all four tracks and should associate two and should not associate the other two. The rule tests objects characteristics conditions during a certain time (5 seconds). Target kinematics represents objects characteristics and time represents the temporal context. Tracks correctly correlated are TP, incorrectly correlated are FP, correctly uncorrelated are TN and incorrectly uncorrelated are FN. The average values taken over 20 (n) executions are shown in Table 2.

Table 2. Case study 1: metrics and results

Scenario	Error $(F_P+ F_N)/n$	Precision $(T_P+ T_N)/n$	Pos. reliability $T_P/(T_P + F_P)$	Neg. reliability $T_N/(T_N + F_N)$	Support T_P / n	Cover $(T_P + F_P) /n$
1	0.01	0.99	1	0.99	0.49	0.49
2	0.02	0.98	1	0.96	0.48	0.48
3	0.01	0.99	1	0.99	0.50	0.50

Low error and high precision values as obtained by the model show that the classifier is working properly, i.e., correlating just tracks that are really supposed to be correlated. High positive reliability values mean that, of all the tracks that have been correlated, almost all of them should indeed have been correlated. The high negative reliability values mean that, from all tracks that have not been correlated, almost all of them were not supposed to be correlated either. The coverage values close to 0.5 combined with high positive reliability values show that there is a balance within correlated tracks and non-correlated tracks. Support values almost equal to coverage values show that almost all tracks that should be correlated have been really done.

4.2 Correlating Tracks with Context Characteristics for Hostility Estimation

A second example considers geographic context. The objective is offers a measure for the hostility of each unknown track. Hostility increases in a scale from 0 to 100. A set of fuzzy rules were created by a domain expert to define the hostility. Rules correlate tracks with commercial routes considering 4 inputs: angle between track´s course and commercial routes; distance from track to route; speed of track; and, the size of track. Fuzzy sets were defined by a domain expert for these 4 variables.

An example of fuzzy rule reflecting the expert belief that the track is probably a freighter may be created relating a track and a route to define the hostility:

IF course closeness IS near AND axis distance IS close AND speed IS regular AND size IS big THEN hostility IS low

Other rules may consider different aspects of context to conclude something about hostility. For instance, an expert can create a rule specifically to detect and classify with high hostility certain targets flying straight to the ship. The rule may be:

IF course closeness IS near AND distance IS close AND size IS small AND speed IS big THEN hostility is severe.

Different fuzzy rules are combined to infer a final hostility for tracks. One test scenario with three tracks was created (fig. 2). First track (T1) is a freighter moving near a commercial route. Second track (T2) is a fighter in a course of interception. The third (T3) is a medium size ship out of commercial routes. Hostility was calculated by rules during the time of test (around 3 minutes). Mean and standard deviation of hostility were calculated for each track. Table 3 shows the results.

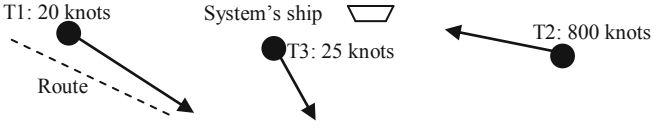


Fig. 2. Tracks used in context hostility estimation (case study 2)

Table 3. Results for case study 2

Track	Hostility Mean	Hostility Standard Deviation
1	20.1	2.1
2	95	0
3	40	0

Hostility varied only for track 1 because this is the only one which is affected by two different rules. Tracks 2 and 3 fired only one rule each, hence the deffuzyfied value corresponds to the maximum of output fuzzy function. Hostility grades produced by model in this test were compliant with those supplied by human experts.

5 Conclusion and Future Works

This work presents an alternative mode to perform context representation and reasoning. The main concern was in the presentation of the hybrid solution adopted and its use in critical contexts. The tests show that the RT-MLR can be applied to real-time applications and that the expressiveness of knowledge domain is determinant for decisions.

As mentioned, the adoption of the event-based approach allows the investigation of fewer rules. Only the rules associated with context fact will be investigated. However, this solution introduces the problems of code maintenance, because the rules are

defined in source code. As future work, a mechanism is being implemented to allow rule management in execution-time. It is also an intention to perform other types of data fusion on different levels, classifying them according to the type of logic and objective. Hence, future works can provide a more thorough comparison of the RB approach with the statistical approach in the field of data fusion.

References

1. Want, R., et al.: The active badge location system. ACM Transactions on Information Systems 10(1), 91–102 (1992)
2. Byun, H.E., Cheverst, K.: Utilizing context history to provide dynamic adaptations. Journal of Applied Artificial Intelligence 18(6), 533–548 (2004)
3. Baader, F., Bauer, A., Baumgartner, P., Cregan, A., Gabaldon, A., Ji, K., Lee, K., Rajaratnam, D., Schwitter, R.: A Novel Architecture for Situation Awareness Systems. In: Proceedings of the 18th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, pp. 77–92. Springer, Heidelberg (2009)
4. Strang, T., Linnhoff-Popien, C.: A Context Modelling Survey. In: 1st Int. Workshop on Advanced Context Modeling, Reasoning and Management (2004)
5. Bar-Shalom, Y., Blair, W.: Multitarget-Multisensor Tracking: Applications and Advances (2000)
6. McCarthy, J.: Notes on formalizing contexts. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp. 555–560. Morgan Kaufmann, San Mateo (1993)
7. McCarthy, J., Buvač, S.: Formalizing context (expanded notes). In: Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural Language, Menlo Park, California, American Association for Artificial Intelligence, pp. 99–135 (1993)
8. Lesaint, D., Mehta, D., O'Sullivan, B., Quesada, L., Wilson, N.: Context-Sensitive Call Control Using Constraints and Rules. In: Cohen, D. (ed.) CP 2010. LNCS, vol. 6308, pp. 583–597. Springer, Heidelberg (2010)
9. Bouzy, B., Cazenave, T.: Using the Object Oriented Paradigm to Model Context in Computer Go. In: Proceedings of Context 1997 (1997)
10. Hong, J., Suh, E., Kim, S.: Context-aware systems: A literature review and classification. Expert Syst. Appl. 36, 8509–8522 (2009)
11. Murayama, T., Kushima, K., Ishigaki, S.: An inference mechanism suited for real-time control. In: Proceedings of the 2nd international conference on Industrial and engineering applications of artificial intelligence and expert systems - Tullahoma, Tennessee, vol. 1, pp. 245–253. ACM, United States (1989)
12. Dörflinger, M.: Interpreting Courteous Logic Programs. Diploma Thesis (2005)
13. Allen, J.F.: An interval-based representation of temporal knowledge. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, vol. 1, pp. 221–226. Morgan Kaufmann Publishers Inc, Vancouver (1981)

DAFNE – A Distributed and Adaptive Fusion Engine

Maarten Ditzel, Sebastiaan van den Broek,
Patrick Hanckmann, and Miranda van Iersel

TNO
Oude Waalsdorperweg 63, 2597 AK The Hague, The Netherlands
maarten.ditzel@tno.nl
http://www.tno.nl

Abstract. An architecture for a distributed and adaptive multi-sensor fusion engine is described. The engine is developed to improve the protection of European armed forces against the threats troops are facing in urban environments. In particular, the engine needs to be scalable, flexible to quickly adjust to different missions and mission goals, easily extendable and suited for distributed implementation. The architecture follows a layered approach to define the main information processing elements.

Keywords: Adaptive distributed fusion, fusion management.

1 Introduction

To improve the protection of European armed forces against threats troops are facing in urban environments, at the end of 2006, the European Defence Agency launched a large research program on Force Protection. The program targets research and technology goals in five capability areas, one of which specifically addresses "data analysis including data fusion from various sources" [1] to improve force survivability by increased situational awareness.

Part of the Force Protection program is the project DAFNE: *Distributed and Adaptive multi-sensor FusioN Engine*. DAFNE aims to demonstrate the advantages of fusing data of multiple heterogeneous sensors combined with intelligence sources over single-sensor operations in an urban environment. Integral part of the project is the development and implementation of a multi-target, multi-sensor and multi-platform fusion engine, capable of mission dependent tracking, classification, situation and threat assessment. This paper focuses on the architecture of the fusion engine, describing the design goals guiding the system design, the individual modules, their roles and their interfacing.

First, Section 2 presents a brief overview of related developments. Then, Section 3 describes the design goals that guide the definition of the engine's architecture. In Section 4 the system architecture is explored. Conclusions about the design and status of the description are summarized in Section 5.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 100–109, 2011.

2 State of the Art

When addressing system architecting for fusion systems, one should take into account several related scientific communities. A few of the most influential are: (i) information fusion, (ii) robotics, (iii) wireless sensor networks (WSN), and (iv) software engineering. Each community has its own application domains, development history, conceptions and terminology.

For instance, the information fusion community originates from the development of algorithms (e.g., tracking, classification) aiming to provide the best possible situation awareness from sensory data, and has a strong background in military applications. On the other hand, the relatively young field of WSN has a strong focus on energy restricted ad-hoc networks, where the optimization of (unreliable) communications to exchange data is a major topic (see for instance [6]). Also concepts and methodologies from software engineering such as design patterns, autonomous multi-agent systems and service oriented architectures (e.g., [5] and [7]) significantly influence the ideas on and the development of fusion architectures.

Architectural efforts in these areas usually concentrate on specific aspects or dedicated applications. To name a few architectural approaches: Stotz addresses a distributed architecture for time-critical target tracking and acquisition [10]. However, it does not take into account high level functionalities such as situation or impact assessment. Laudy on the other hand, does address situation assessment [4], but pay little attention to other functionalities. In [11], the authors present a service oriented architecture for data acquisition and information, specifically tailored towards sensor networks consisting of small devices with limited processing and communication capabilities. Zug and Kaiser describe a distributed architecture primarily aimed to efficiently exchange incidentally faulty data over unreliable channels [12].

This article primarily takes a functional approach based on information fusion concepts. In this domain, several fusion process models have been developed over the years. Some of them stand out as classic references for the information fusion scientific community. The first and best known model originates from the US Joint Directors of Laboratories (JDL) [2] in 1985.

The JDL model comprises five levels of data processing and a database, which are all interconnected by a bus. The five levels are not meant to be processed in a strict sequential order and can also be executed concurrently. Later, Steinberg and Bowman proposed several revisions and expansions of the JDL model ([8] and [9]).

A recent revision of the JDL model is described in [3]. The proposed Networked Adaptive Interactive Hybrid Systems (NAIHS) model is closely related to the JDL model with respect to the data abstraction levels of the data fusion. NAIHS uses:

- *Signal level,* providing estimation of states of sub-object entities (e.g., signals, features, data);
- *Object level,* providing estimation of states of discrete physical entities (e.g., vehicles, buildings);

- *Situation level,* providing estimation of relationships among entities (e.g., grouping, cueing, acting on); and
- *Impact level,* providing estimation of impacts (e.g., consequences of threat activities on ones own assets and goals).

Furthermore, the NAIHS model distinguishes three modelling principles: (i) information abstraction, (ii) time horizon of goals and (iii) physical structure of the system in its environment. The architecture derived in this paper builds on these three principles.

- *Information Abstraction*: The NAIHS model has four different levels of information abstraction which are all coupled to an assessment and management function. Based on the generated situation awareness, an intention is generated at each level. The combined intentions lead to signals which are sent to the effectors.
- *Time Horizon*: Every decision making or management process, reasons about possible outcomes in the future taking into account possible actions. The time horizon of these processes may have a wide time range. Therefore, a decomposition of the process cycle in this dimension has been adopted in various application domains. In the military field a strategic, operational and tactical level is in use.
- *Physical Structure*: A third principle on which a decomposition can be based is a physical structure. In systems considered here, the physical structure includes a network of platforms containing various collectors and effectors, and the possibility for the assessment and management components to be physically separated.

The NAIHS model uses a hierarchical structure to model the Information Abstraction. The Physical Structure can be depicted as a third dimension in the otherwise two dimensional hierarchical structure representing the instantiations of the different physical structures.

3 Design Goals

The DAFNE project aims to design an experimental distributed multi-sensor fusion engine that will combine data from heterogeneous sensors in urban warfare scenarios to enhance situational awareness during military operations. This requires the generation of reliable estimates about entities and events, their relations and possible impact in the observed areas. Supported sensors and platforms are very diverse and may exhibit very different characteristics.

The heterogeneous sensors provide data that can be exploited for tasks such as target detection, localization, tracking, identification and recognition. Each task can benefit from the availability of multiple sources providing additional data of the same or complementary type, thus adding redundant or richer information. In both cases the reliability of the estimates on a particular physical entity can be improved through fusion processes that fall under levels 1 and 2 of the JDL model. The improved confidence will support a decision making process.

For this, a versatile fusion architecture is needed. The architecting process is governed by the following design goals:

- *Scalability*
  The fusion engine shall be able to operate with a varying and heterogeneous set of input sensors, allowing the engine to scale in accordance with available equipment in different missions, e.g., radar, seismic, and optical sensors.
- *Flexibility*
  The fusion engine shall be able to operate in different (urban) environments without the need to make major modifications to the system. The engine will utilize mission dependent contextual information to improve its performance, e.g., by making use of terrain data to manage, select and filter sensor data.
- *Extendibility*
  The fusion engine shall be easily extended for future needs, i.e., it shall be future proof, for instance to accommodate new classification mechanisms, kinematic models, and also dynamic situation and threat assessment techniques (corresponding to the levels 2 and 3 of the JDL model).
- *Distributedness*
  The fusion engine shall be able to operate in a distributed manner. Multiple instances shall be able to share their local intermediate results to come to a better global awareness.

Given the above, a highly configurable, loosely coupled fusion engine is foreseen. The system is highly configurable to enable quick and easy tuning of the engine to a changing mission environment and needs. It is loosely coupled to enable distributed instantiations and to ease extension and maintenance of the engine.

4 System Architecture

In this section a number of aspects of the DAFNE system are described starting with the system context. Next, the main functional flow of the DAFNE system is described, followed by a description of the main elements. These elements form the Fusion Engine, which encompasses the forward flow of information through the system, and the Fusion Management, which describes how the Fusion Engine can be managed based on higher level demands.

4.1 System Context and Use

Any system, including the DAFNE system, will always operate in an enclosing environment interacting with the system itself. Therefore, any architectural attempt has to start with the definition of the system's context, defining the system boundaries and the scope of the system to be designed.

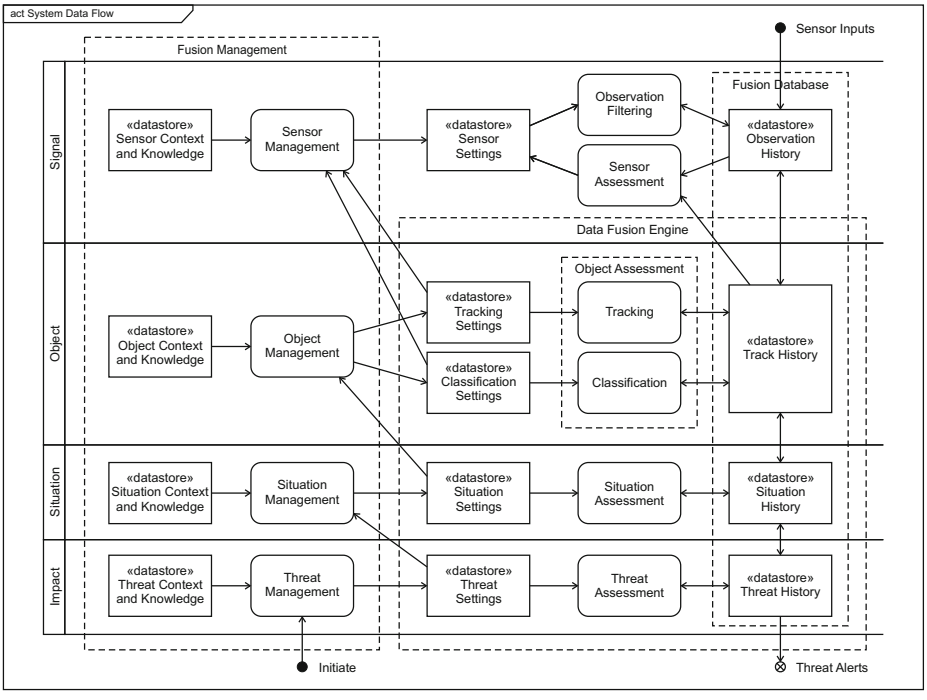


Fig. 1. Data flow diagram of the DAFNE fusion engine

Apart from the DAFNE system itself, two other systems are identified: a Sensor Simulator and a Performance Evaluator. In the DAFNE project, these are added to provide stimuli and to assess the performance of the fusion engine. In reality, input would be sensor data from many systems, and output would be assessed threats. Knowledge about the environment in which the system operates needs to be available to the system. The context information in the DAFNE system, describes relevant parts of the environment, such as geographical information and databases or rules for classification and definition of threats. A part of this context information is mission independent (e.g., movement models of a pedestrian), and a part is mission dependent (e.g., a map of the city).

A scenario in which a system based on the proposed architecture can show its relevance is situated in an urban warfare situation. A terrorist attack with a car bomb is planned on an embassy located in a city. An automated system using input from many sensors will raise an alarm when a suspicious situation occurs. In this example, this suspicious situation consists of finding an unkown car that stops near the embassy, which has a relation to an area of the city where car bombs are known to be produced, after human intell indicated a possible planned attack. Context information may change if the system is deployed in another city, if threats change to other types of vehicles, or if suspected cars are expected to have a different appearance.

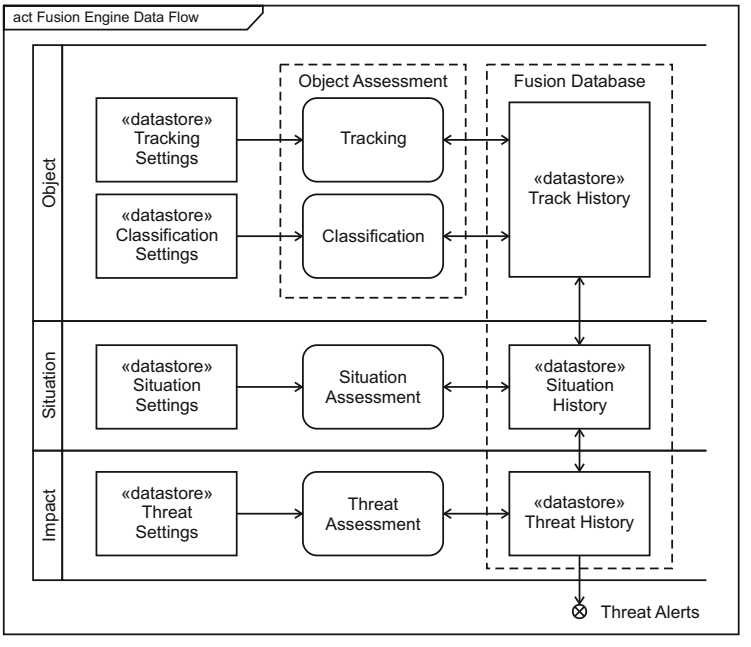


Fig. 2. Partial view of the DAFNE system data flow focussing on the Fusion Engine

4.2 Functional Design

Figure 1 shows a SysML data flow diagram, depicting the functional decomposition of the DAFNE Fusion Engine and its surroundings. The identified functionality is allocated and decomposed according to the four main NAIHS levels:

- Signal assessment and management (level 0),
- Object assessment and management (level 1),
- Situation assessment and management (level 2),
- Impact assessment and management (level 3).

The data flow in the DAFNE system will for a large part be through data stores. A module obtains new data from a data store, updates and adds information, and puts the result back, where it can be retrieved for further processing. All modules that combine and process sensor information (resulting in assessed threats) are grouped into the Fusion Engine.

4.3 Fusion Engine

The most important processes in the data-flow of the DAFNE System are tracking, classification, situation assessment, and threat assessment. This partial view of the data flow diagram is given in Figure 2, which zooms in on the functionality of the Data Fusion Engine.

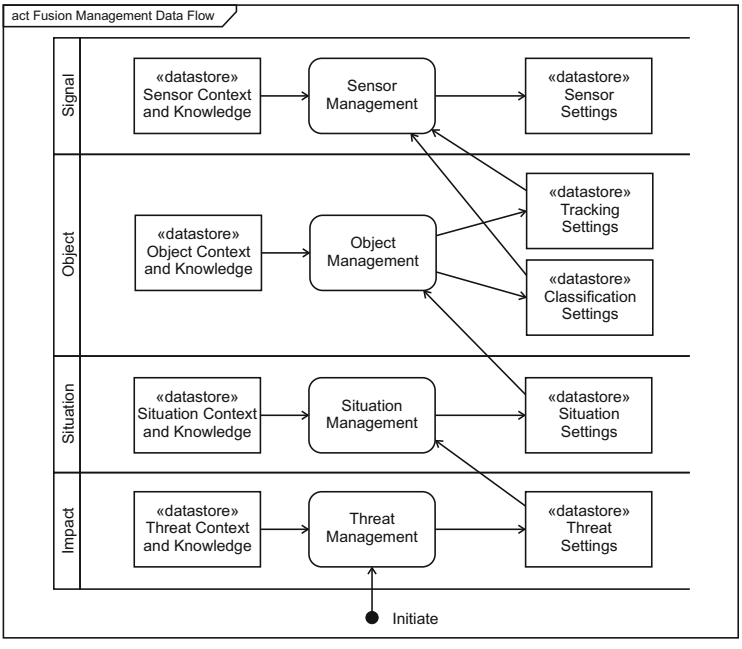


Fig. 3. Partial view of the DAFNE system data flow focussing on the Fusion Management

As the figure shows, data from different sensors is processed by a Tracking component. This data may be tracks produced by a sensor or single measurements. The main task of the Tracking component is associating information from different sensors to single objects. This does not only include the kinematic data (which has to be updated using new measurements), but also measured features, labels or classes that a single sensor might already have determined individually. From these measurements, other labels can be derived (for example, temporal analysis provides the label 'Stopping'), different features can be combined (e.g., classification by pattern recognition) or labels from classification processes in sensors can be combined. In the latter case, an important objective is to handle contradicting information of sensors.

Aggregation from features and labels to classes is performed by the Classification component. Taking into account the information provided by the sensors and the Tracking component, the Classification component tries to identify the class to which an object belongs. The requirement is that new individual objects are placed into groups based on quantitative information on one or more measurements, and based on a training set or rule set in which previously decided groupings are already established. Since sensors may have already provided a (partial) classification, another important function of the classification step is disambiguation of contradicting evidence.

The Situation Assessment component determines relations between objects. This can include relations between different object tracks (for instance, to determine that a person met another person) and linking tracked objects to objects from the contextual information (e.g., a car is in a certain part of a city). Outputs of this component are tracks of objects, with relations to other objects.

All information defining an object and relations to other objects (now or in the past), are evaluated by the Threat Assessment component. Each threat is defined as a rule, linking the threat to situations that are assessed in Situation Assessment. Rules can for example described by a Baseyan belief network. A rule could be: if belief is high enough in an object having class car, and the object having label stopping and the object being near the embassy, give an alert.

Throughout the processing, an object (tracked or from context) is defined by its state, described by values or labels for different features, classes and situations. Each module in the fusion engine updates this state, making it more accurate or more complete. In the object layer, this is done by the Tracking and Classification components adding more measurements of the same feature (e.g., location from radar or electro-optical sensors), by associating complementary information from sensors (e.g., color from a camera to a radar track) or by deriving classes (e.g., from velocity or classification from other features). In the situation layer, relations are added by the Situation Assessment component using distances in position, and time. In the impact layer, the Threat Assessment component assesses the threat for all tracked objects, based on all propositions (labels, classes, relations, and contextual information).

4.4 Fusion Management

As shown in the diagram in Figure 3, the fusion management works in the opposite direction from the DAFNE Fusion Engine. In general, the management modules translate the knowledge of the outside world to information relevant to the fusion engine functional blocks, based on the demands of higher level modules. The environment is available as external mission dependent and independent context information.

Threat Management selects and defines the relevant threats, for example as rules classifying situations as suspicious or threatening (for instance, a suspicious car stops in front of an embassy). Similarly, Situation Management selects and defines the relevant situations (e.g., a car coming from a suspicious area, and cars slowing down near the embassy). In this case, relevancy may be defined by the definition of threats as provided by the Threat Management to Threat Assessment.

Again similarly, Object Management selects and defines the relevant objects (e.g.detecting cars, detecting deceleration), steering the classification and the tracking processes. Finally, given the desired objects, Sensor Management selects and configures the available sensors. By this cascading process, Fusion Management incorporates changes on the top level to automatically adjust all settings and contextual information accordingly and appropriately on all levels.

5 Discussion and Conclusions

The presented design of the DAFNE system provides a generalized description of a fusion system that can be used for implementation of the system, providing fusion of multiple sensors for situation and threat assessment in an urban environment. The architecture allows for easy reconfigurability for use in other applications or environments, by changing the contextual information, and settings that control the different modules. Since interfacing of modules is mainly defined by what level of information is going in and out, replacing or extending processing is possible as well, without the need to to change other parts of the system. The design fulfils the design goals given in Section 3, as discussed below.

Concerning scalability, the system does not put requirements on the sensor input, other than that there are features linked to the detected information, that can be used to associate with other sensors. Generally, this will be kinematic data, but data could associate on other features, when only a rough location (of the sensor) is known. Communication is done via a generic data store that can efficiently handle data from asynchronous and/or geographically separated sensors. The tracker module is also able to process tracks provided by other DAFNE nodes, associating in a similar way to sensor data. This allows for distributing the data load, if the number of sensors increases, for example by having tracker modules process data of sensors that are geographically close, and combining those tracks in a different DAFNE node.

With respect to flexibility goal, the different engine modules use contextual information (such as geographical objects, classification and threat assessment data) as a description of the system's operational environment. This information can be updated for a new environment, for example to include locations of geographical objects, specific classification rules for objects that look differently in a new location, or to include statistical information about the occurrence of certain objects.

The Fusion Management allows the functional modules to generate information that is relevant to higher level modules, by translating contextual knowledge to information in the form the module requires. For example, a classification module could get information which features are relevant to classify a certain object, based on whether situation and threat assessment are interested in such objects, and on how these features describe such an object in the current environment. In this way, changes at a high level, for example a user indicating an interest in different threats, would automatically change contextual input of all lower level modules.

No limitations are put on the functionality of the modules by the architecture. Functionality can be extended by adding more algorithms, and by providing more or better contextual information, satisfying the extendibility goal. For example, when a tracker is only using radars as sensors then a radar-optimised tracking algorithm can be used, while a tracker with triangulation support can be used when optical sensors are included in the sensor suite.

Finally, concerning distributed implementation, the output of a DAFNE node (containing the Fusion Engine and Fusion Management) can be used as input by

other nodes at different levels, by treating it as a sensor that provides already tracked objects. As described above, for scalability, this allows different sensors to provide data to different nodes. It also allows for different levels of processing to run on different nodes, for example doing situation and threat assessment in a centralized location, combining classified tracks from different geographical areas.

Acknowledgements

This work is done as part of the DAFNE project and is sponsored by the European Defence Agency (EDA) under grant number A-0380-RT-GC. The authors would like to thank the partners in the DAFNE consortium for their valuable comments and contributions regarding the architecture of the fusion engine.

References

1. European Defence Agency. EDA defence R&T joint investment programme on force protection (JIP-FP), http://www.eda.europa.eu/genericitem.aspx?id=184
2. Hall, D.L., Llinas, J.: An introduction to multisensor data fusion. Proceedings of the IEEE 85, 6–23 (1997)
3. Kester, L.: Method for Designing Networking Adaptive Interactive Hybrid Systems. In: Babuska, R., Groen, F.C.A. (eds.) Interactive Collaborative Information Systems. SCI, vol. 281, pp. 401–421. Springer, Heidelberg (2010)
4. Laudy, C., Petersson, H., Sandkuhl, K.: Architecture of Knowledge Fusion within an Integrated Mobile Security Kit. In: 13th Conference on Information Fusion (2010)
5. Lu, C.H., Wu, C.L., Fu, L.C.: A reciprocal and extensible architecture for multiple-target tracking in a smart home. IEEE Transactions on Systems, Man, and Cybernetics. Part C: Applications and Reviews 41(1), 120–129 (2011)
6. Luo, H., Tao, H., Ma, H., Das, S.K.: Data Fusion with Desired Reliability in Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems 22 (3), 501–513 (2011)
7. Rothenhaus, K., Michael, J., Shing, M.T.: Architectural patterns and auto-fusion process for automated multisensor fusion in soa system-of-systems. IEEE Systems Journal 3(3), 304–316 (2009)
8. Steinberg, N., Bowman, C.: Revisions to the JDL data fusion process model. In: Proceedings of the 1999 National Symposium on Sensor Data Fusion (May 1999)
9. Steinberg, N., Bowman, C.: Rethinking the JDL Data Fusion Levels. In: Proceedings of the National Symposium on Sensor Data Fusion JHUAPL (June 2004)
10. Stotz, A., Sudit, M.: Information fusion engine for real-time decision-making (INFERD): A perceptual system for cyber attack tracking. In: 10th International Conference on Information Fusion. IEEE, Los Alamitos (2007)
11. Tapia, D.I., Rodríguez, S., Bajo, J., Corchado, M., Garcia, O.: Wireless sensor networks for data acquisition and information fusion: A case study. In: 13th Conference on Information Fusion (2010)
12. Zug, S.D., Kaiser, J.A.: An architecture for a dependable distributed sensor system. IEEE Transactions on Instrumentation and Measurement 60(2), 408–419 (2011)

Context Representation and Fusion via Likelihood Masks for Target Tracking

Lauro Snidaro, Ingrid Visentini, and Gian Luca Foresti

Department of Mathematics and Computer Science
University of Udine, Italy
lauro.snidaro@uniud.it

Abstract. The inclusion of contextual information in low-level fusion processes is a promising research direction still scarcely explored. In this paper we propose a framework for integrating contextual knowledge in a fusion process in order to improve the estimation of a target's state. In particular, we will describe how contextual information can take the form of likelihood maps to be fused with the sensor's likelihood function in order to encode the dependence of the observation from the context.

Keywords: data fusion, contextual information, likelihood masks.

1 Introduction

The analysis and study of contextual information and its relations to an entity of interest has a long history in many and diverse research fields: from Linguistics to Cognitive Psychology to Artificial Intelligence [14]. Major challenges related to context are generally:

- how to formally define context
- how to represent context
- how to discover and reason about relevant contexts for particular applications.

All the information related to the scenario and the situation in which the observed entities are can be considered contextual information. In the literature, many consider only position, surroundings, identity and time as context, with some extensions to the status and the applications [9]. Furthermore, there are very general definitions where context is a subset of a physical or conceptual state status, which is related to a specific entity [3]. A definition can be drawn from [3], where the context is formalized for a generic software application as "any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application". More in general, in our case, this proposition can be rewritten as

"Context is any information that can be used to characterize the situation of entities that are considered relevant to the interaction between the operator and the system"

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 110–117, 2011.

In a different application scenario [4,2] three different elements are identified as significant to context: places, people and things (in the original intention, they refer to physical or software components). In [6] the authors introduce contexts as abstract mathematical entities in a more general framework which includes context-sensitivity, namely knowledge represented by contextual information systems. More in general, the contextual information has been previously incorporated into levels, under the umbrella of Bayesian framework and the name of relaxation labeling [13].

As it can be seen, context is a precious source of information in every automatic system for situation awareness. Contextual information can play different roles at different levels, providing significant cues that can range from ancillary data to semantically rich goal-related information. In this paper, we will address the problem of defining and representing context in a data fusion application in order to improve the estimation of a target's state, as discussed in the following sections.

2 Contextual Information in Data Fusion

Context awareness allows the possibility for a system to change and adapt to better react to unpredictable or potentially harmful events. The system should not only adapt to changes in the availability of resources, but also to the presence of new or updated contextual information, such as time of the day or exploiting differently the information a sensor has about that object being observed. This form of adaptation is pertinent to JDL model Level 4 [5] optimization schemes and policies the system may have to continually optimize its performance [11].

As already observed in [14], context may be exploited in data fusion system at different levels. Up to now, its most exploited use is to participate in inferences on the current or future situation [14]. Contextual knowledge provides in fact a powerful way to semantically bind sensor measurements and real-world observables. This is particularly evident for surveillance systems where, for example, the measured position of a target is checked against the location of sensible areas possibly triggering an alarm if a relevant "suspicious" event is recognized (e.g. "if target's position is inside forbidden area, then warn operator") [10]. According to the JDL model, this kind of inferences belong to Levels 2 and 3.

In data fusion problems, multiple measurements of a given observable are filtered to refine an estimate of it. In this case, context provides a description of the surrounding environment, and helps in handling and reducing the ambiguity/uncertainty of detected measurements or events [7] thus being functional in *refining* filtered estimates of the measurements. Therefore, it provides essential information for sensor fusion. For instance, the information "today it is raining" can help to better model a situation when only electro-optical sensors are working, or exploit a subset of focused sensors when "the target is at location (x, y, z)". In terms of JDL model, this exploitation of contextual information belongs to Level 1 processing and can be used to [14]:

– refine ambiguous estimates;
– explain observations.

The application of information derived from context to Level 1 processing is still under research and only a few attempts have been made. For example, in [8] a tracker supervised by contextual information was proposed. Essentially, the author proposes to improve tracking performance by selecting a subset of all the available sensors (sensor selection) as reliable sources of measurements for a given target. The Kalman update equations are also modified to take into account the probability of "validity" of the sensors thus weighting their contribution in updating the current target's state estimate. The validity of a given sensor is determined by inspecting a vector of contextual parameters defined a priori from human knowledge. These parameters thus define a validity domain for a given sensor under certain conditions. The paper stressed, more than ten years ago, the importance of taking into consideration contextual information in the fusion process. In particular, the work showed how a tracking process (performed by Kalman filtering) could benefit from an appropriate weighting of the observations performed by the available sensors. However, it takes a general approach in considering the contextual parameters vector as composed by an arbitrary number of components, which express contextual (i.e. environmental) and sensory variables. The validity domain of a sensor is defined a priori by a human expert and encoded in the contextual parameters via fuzzy logic.

In [12] the problem of tracking ground targets is addressed by considering topographic variance as contextual information that can influence targets' patterns. In particular, topographical properties of the area are fused with measured data to improve tracking performance of ground targets for example by changing the the variance of the process noise at expected turning points such as road junctions.

Attempts to include contextual information of the observed area in the position estimation process can be found in indoor location systems. An example can be found in [1] where fuzzy logic is used to model the probability of a target's position when moving in a grid, taking into account the environmental map. However, the work does not mention how other type of contextual knowledge could be integrated.

3 Formulation

Contextual information can be a key factor in determining the state of an entity of interest. Context is a form of knowledge that can dramatically impact on the reliability of an observation. Imagine the case of a target moving along a city street and suppose that we want to estimate its state $\mathbf{x}$ as the vector of Cartesian bidimensional coordinates. Suppose now that the observation $\mathbf{z}_t$ at time t is checked against an urban map of the monitored area resolving $\mathbf{z}_t$ as falling inside a building. Now, given the fact that we know that the sensor has no see-through-walls capability, this could be explained as an occasional quirk of the sensor and could be easily filtered out by the tracking algorithm (e.g. Kalman

filter, particle filter,etc). Especially if $\mathbf{z}_t$ resolves inside a building while both the previous state $\mathbf{x}_{t-1}$ and the next measurement $\mathbf{z}_{t+1}$ do not. Unfortunately, in real-world monitoring applications it often happens that a sensor provides a sequence of unreliable observations due to partial occlusion of the target, unfavorable weather conditions, sun blinding, persistent reflections, etc. In these cases tracking can be severely disrupted providing an unreliable estimate of the target's position and trajectory. Checking the measurements against a map of the monitored area is a form of contextual knowledge inclusion that could, as in the latter example, provide an insight on the reliability of the sensor in a specific situation. Knowing the sensing capabilities of a sensor is another form of contextual knowledge that could be exploited conveniently. In the previous example, knowing that the sensor has no see-through-walls capability *and* the fact that the last few measurements fall inside a building can help us in concluding that those measurements may be affected by a form of bias and thus be unreliable. The sensor may be in fact persistently experiencing one or more of the disturbing conditions mentioned above.

This could be further exploited in a multi-sensor system where the same target is observed by multiple sensors. In this case contextual information could be used to weight the measurements provided by the sensors in the fusion process. In a situation as the one describe above, the measurements from the unreliable sensor would be weighted less, thus affecting less the final fused estimate of the target's position.

3.1 Likelihood Masks

Here we propose a way for formalizing and encoding contextual information for target tracking purposes. First we will describe the single sensor case where we want to estimate the state $\mathbf{x}$ of a target understood as its position in the monitored area. More specifically, in a Bayesian framework, we want to find the posterior distribution $p(\mathbf{x}|\mathbf{z}, c)$ of the state, given observation $\mathbf{z}$ and context c. Applying the Bayes theorem, the posterior can be computed as follows:

$$p(\mathbf{x}|\mathbf{z}, \mathbf{c}) = \frac{p(\mathbf{x}|\mathbf{c})p(\mathbf{z}|\mathbf{x}, \mathbf{c})}{p(\mathbf{z}|\mathbf{c})} \tag{1}$$

where $p(\mathbf{x}|c)$ is the prior on the state (e.g. the previous known state), $p(\mathbf{z}|\mathbf{x}, c)$ is the likelihood of observation $\mathbf{z}$ given the state of the target and the context, and $p(\mathbf{z}|c)$ is a normalizing factor.

The interesting term in the equation above is the likelihood $p(\mathbf{z}|\mathbf{x}, c)$ that defines the sensor model. This term is a function of all the variables involved and encodes the measurement model of the sensor. The model for the sensor can be built by fixing the state $\mathbf{x}$ and observing the distribution over $\mathbf{z}$. Here we will assume the sensor model to be Gaussian and we will investigate how to include context in the likelihood function.

Context will take here the form of *likelihood masks* to be fused with the measurement likelihood. Figure 1 shows an example of buildings context likelihood

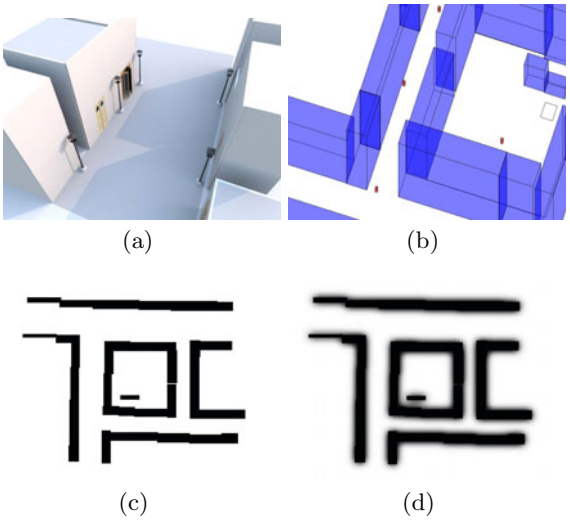


Fig. 1. Example of buildings context: generation of likelihood masks for an electro-optical sensor. (a) Detailed map of an urban area, detail of full 3D model is shown. (b) Buildings are highlighted and selected on the map. (c) 2D top-view map of the buildings (detail of the full map shown). (d) Buildings context likelihood mask (detail of the full map shown) for vehicles.

mask generation for an electro-optical (EO) sensor. From a detailed map of an urban area (a), here a full 3D model is shown, only the buildings are selected (b), a 2D top-view map of the buildings is obtained (c) (detail of the full map shown). Black pixels represent locations where the likelihood of the target's presence is impossible (coinciding with the buildings' extent). On the contrary, white pixels indicate possible locations for the target (in this case the streets). The example mask shown in (c) could be functional in determining possible and impossible locations for a pedestrian target detected by an electro-optical sensor. The separation between possible and impossible locations is sharply defined. Figure 1(d) depicts the building mask that can be used for tracking a vehicle. The map carries the knowledge that a vehicle, contrarily to a pedestrian, is unlikely to be very close to a building and the transition between buildings and streets is therefore blurred to encode this (improbable) possibility.

Figure 1 shows how a building contextual mask for a specific type of sensor (EO) can be generated. The mask encodes the capability of the sensor of detecting targets only in open areas, masking the areas covered by buildings. The example in the figure depicts also the possibility of generating different building maps in relation to the type of target. In general, for tracking purposes, the vast majority of contextual information can be encoded in the form of likelihood masks. The idea is that the detection capabilities of the sensor can be influenced by context in many different ways and that a mask can be maintained for every type of "disturbance". The drawing in Figure 2 depicts possible contexts that can be influential to the final target's location estimate. Figure 2 (a) shows the mask

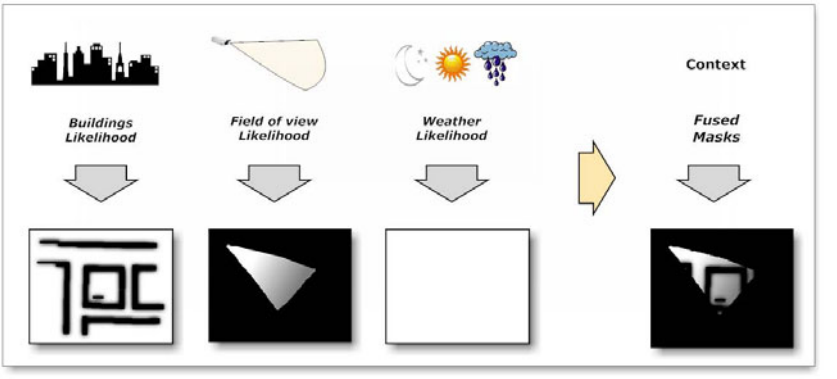


Fig. 2. Example of likelihood masks for different contextual information. (a) buildings contextual mask, (b) field of view mask, (c) weather mask, (d) fused contextual mask.

related to the effects of buildings and structures to the observation capability of the sensor (as in Figure 1). The mask in (b) indicates the field of view of the sensor thus excluding the possibility of a detection outside its range. (c) weather mask, the example shows the case of non-disruptive conditions, as a result the mask is completely white. The fusion of the three contextual masks yields the one shown in (d). The masks illustrated here are merely exemplar and others could be developed depending on the application. All contextual masks are in principle dynamic and could be updated as soon as new information is available. This could be done manually by an operator or automatically depending on the type of context: the update of the buildings mask would probably need some kind of human intervention (e.g. a building is collapsed or destroyed); the field of view map could be recalculated automatically rather easily in the case of Pan-Tilt-Zoom cameras or in the case of sensor relocation; the weather map could also be updated automatically processing weather radar maps. However, the updating of likelihood masks is out of the scope of the present paper and it will be investigated in future research.

3.2 Likelihood Fusion

As regards to implementation details, the masks are here conceived as 2D matrices with values in the $[0, 1]$ interval. The actual fusion of all the l contextual masks is performed by Hadamard product (entrywise multiplication), that is:

$$C = C_1 \circ C_2 \circ \ldots \circ C_l \tag{2}$$

where C is the fused mask and the Hadamard product of two $m \times n$ matrices A and B is defined by $[A \circ B]_{ij} = [A]_{ij}[B]ij$ for all $1 \leq i \leq m$, $1 \leq j \leq n$.

The fused mask C is then used in the likelihood function $p(\mathbf{z}|\mathbf{x}, \mathbf{c})$ as follows:

$$p(\mathbf{z}|\mathbf{x}, \mathbf{c}) = p(\mathbf{z}|\mathbf{x})[C]_{\mathbf{x}} \tag{3}$$

(a) (b) (c)

Fig. 3. (a) Fused contextual masks, (b) observation likelihood map, (c) final likelihood map

Figure 3 shows the application of (3): (a) is the fused mask shown in Figure 2, (b) is the observation likelihood, and (c) is the final observation likelihood incorporating all available contextual information. In the example above, the observation likelihood (b) is multimodal, meaning that two possible targets have been detected. This may be due to the actual presence of two targets or it could be that one or both are false alarms given by poor detections (e.g. noise, reflections, etc.). It is interesting to notice that the detection on the right (brighter Gaussian) appears to be stronger[1] than the one on the left, indicating higher likelihood of target's presence. The integration of contextual knowledge as per (3) overturns the likelihoods of the two detections as it can be seen in the fused likelihood (c). The detection on the right was resolved as falling inside a building (probably a reflection) and masked by contextual information. The mask modeling the sensor's field of view is also a decreasing gradient as shown in Figure 2. This was done to model the decreasing capability of the EO sensor to detect far away targets. In the above example, the inclusion of contextual knowledge has favored the likelihood of the detection closer to sensor and compatible with the occupancy of the area.

4 Conclusions

We have explored a framework for integrating contextual knowledge in a Bayesian fusion process for target's state estimation. In particular, we have presented a way of modelling context as a set of masks that can be easily integrated with sensor observation likelihoods for target tracking purposes. Each mask is intended as a way of constraining the detection capability of a sensor according to knowledge relative to the observed environment. Future work will be directed to the analysis of the multi-sensor case and the implementation in a real-world scenario.

[1] Detection *quality* in the case of an EO sensor could be computed according to, for example, the confidence (or quality) value produced by a tracking-by-classification algorithm [10].

Acknowledgments

This work was supported by the European Defence Agency within the DAFNE (Distributed and Adaptive multisensor FusioN Engine) project A-0830-RT-GC.

References

1. Bohn, J., Vogt, H.: Robust probabilistic positioning based on high-level sensor-fusion and map knowledge. Tech. rep., ETH, Zurich, Switzerland (2003)
2. Dey, A., Abowd, G.: Cyberdesk, the use of perception in context-aware computing. In: Proceedings of the 1997 Workshop on Perceptual User Interfaces, pp. 26–27 (October 1997)
3. Dey, A., Abowd, G.: Towards a better understanding of context and context-awareness. In: Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, New York, USA (2000)
4. Dey, A., Salber, D., Abowd, G.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction Journal 16(2-4), 97–166 (2001)
5. Llinas, J., Bowman, C.L., Rogova, G.L., Steinberg, A.N., Waltz, E.L., White, F.E.: Revisiting the JDL data fusion model II. In: Svensson, P., Schubert, J. (eds.) Proceedings of the Seventh International Conference on Information Fusion, vol. II, pp. 1218–1230. International Society of Information Fusion, Stockholm, Sweden (June 2004), http://www.fusion2004.foi.se/papers/IF04-1218.pdf
6. Martín-Vide, C., Mitrana, V.: Contextual information systems. In: Dey, A.K., Kokinov, B., Leake, D.B., Turner, R. (eds.) CONTEXT 2005. LNCS (LNAI), vol. 3554, pp. 304–315. Springer, Heidelberg (2005)
7. Murphy, R.R.: Biological and cognitive foundations of intelligent sensor fusion. IEEE Transactions on Systems, Man and Cybernetics 26(1), 42–51 (1996)
8. Nimier, V.: Supervised multisensor tracking algorithm. In: International Conference on Multisource-Multisensor Information Fusion, Las Vegas, USA, pp. 149–156 (1998)
9. Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., Van de Velde, W.: Advanced interaction in context. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 89–101. Springer, Heidelberg (1999)
10. Snidaro, L., Visentini, I., Foresti, G.L.: Quality based multi-sensor fusion for object detection in video-surveillance. In: Intelligent Video Surveillance: Systems and Technology, pp. 363–388. CRC Press, Boca Raton (2009)
11. Snidaro, L., Visentini, I., Foresti, G.L.: Sensor management: A new paradigm for automatic video surveillance. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 452–459. Springer, Heidelberg (2010)
12. Sodtke, C., Llinas, J.: Terrain based tracking using position sensors. In: International Conference on Information Fusion, vol. 2, pp. 27–32 (2001)
13. Song, X.B., Abu-Mostafa, Y.S., Sill, J., Kasdan, H., Pavel, M.: Robust image recognition by fusion of contextual information. Information Fusion 3(4), 277–287 (2002)
14. Steinberg, A., Rogova, G.: Situation and context in data fusion and natural language understanding. In: Proceedings of the Eleventh International Conference on Information Fusion, Cologne, Germany (2008)

Adaptive Data Fusion for Air Traffic Control Surveillance

Juan A. Besada, Guillermo Frontera, Ana M. Bernardos, and Gonzalo de Miguel

Universidad Politécnica de Madrid
Spain
besada@grpss.ssr.upm.es

Abstract. This paper describes a method to enhance current surveillance systems used in air traffic control. Those systems are currently based on statistical data fusion, relying on a set of statistical models and assumptions. The proposed method allows for the on-line calibration of those models and enhanced detection of non-ideal situations, increasing surveillance products integrity. It is based on the definition of a set of observables from the fusion process and a rule based expert system with the objective to change processing order, algorithms or even remove some sensor data from the processing chain.

Keywords: Data Fusion, Surveillance, Integrity assessment, Expert systems.

1 Introduction

Multisensor multitarget tracking (MMTT) systems are the basis of modern Air Traffic Control surveillance systems. They rely on the coupled operation of:

- Association systems, obtaining a unique target track from all data received from all sensors. They contain means for track initiation, association of measures to tracks, and track deletion for tracks not receiving updates.
- Highly accurate tracking filters (such as IMM filters [1]), which exploit all available sensor measurements enabling extremely fast manoeuvre detection (note IMM filters perform a kind of probabilistic assessment of model compliance which can be understood as a manoeuvre detection).

In order to be computationally efficient, zero mean Gaussian measurement error statistics, with known variances, are usually assumed, and therefore any mismatch between this model and actual measurements would lead to either association or manoeuvre detection problems and therefore to reduced quality of tracking results.

Due to this problem, much effort has been devoted in the last years to the definition of bias estimation procedures for multisensor multitarget tracking systems (among others [2][3][4]). The basic idea is estimating all bias terms in the measurements potentially causing consistency mismatch, and removing them from the raw measures, providing the tracking filters with bias corrected (therefore unbiased) measures.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 118–126, 2011.

These efforts have been mainly concentrated to radar bias estimation and correction, as they were the most widely used sensors in Air Traffic Control environments. A modern Air Traffic Control Data Fusion system must be able to process at least the following kinds of data:

- Radar data, from primary (PSR), secondary (SSR), and Mode S radars, including enhanced surveillance.
- Multilateration data from Wide Area Multilateration (WAM) sensors.
- Automatic dependent surveillance (ADS-B) data.

The complementarity nature of these sensor techniques allows a number of benefits (high degree of accuracy, extended coverage, enhancements to systematic errors estimation and correction, etc.) and brings new challenges for the fusion process in order to guarantee an improvement with respect to any of the sensor techniques used alone. The fusion of all measurements requires new solutions and a robust process that considers detailed characteristics of all data sources and checks their consistency before being fused.

In this paper we extend previous works to define a novel approach to cover bias estimation, noise covariance estimation (present on some systems), association and detection estimation and real time adaptation to current sensor situation.

This kind of adaptation is especially important for ADS-B data, where the information is provided by on-board navigation systems whose integrity is not guaranteed to the same level as centralized or distributed sensors as radars or WAM.

Non-cooperative sensors (PSR) may suffer from the presence of false alarms due to thermal noise or reflections on ground, sea or rain (clutter). By other processes (i.e. fruit, multipath reflections, etc.) false alarms may also appear in SSR or WAM sensors. In addition, all sensors have a detection characteristic: not every attempt to perform detection is successful with radar sensors, and reception of ADS-B and WAM data does not have the nominal measurement rate but a reduced one.

In general, in all considered measurement sources there are two types of sources of error:

- Random terms (i.e. thermal noise or measurement timestamp jitter), usually modelled as white noise.
- Constant or slow-changing terms, spatially correlated, which may be modelled as bias.

Additionally, bias terms can be divided in several subtypes:

- Sensor related bias, which have a same value independently of the target.
- Target related bias, which are equal for all sensors of the same type, independently of the sensor.

In addition, a certain percentage of measurements suffer errors much in excess of nominal statistics. They are usually marked as outliers, and they are due to some non-linear effect of the sensor, probably due to some kind of malfunction.

In stationary or slow changing conditions, all those problems' effect on tracking may be alleviated through manual adaptation, tuning the association and filtering processes to the current sensor situation. If there is a sudden change in detection or error behaviour of a given sensor, the system must be able to respond rapidly (and

therefore automatically) in order to either adapt to the new conditions, or to even preclude the use of some sensor data.

The paper starts with the definition of all sensors of interest detection and error models, and then describes and justifies the overall sensor state assessment process. Then, the adaptation procedure is described, and a simulation-based example for some of the described problematic situations is included.

2 Sensor Measurement Process Description

2.1 ATC Radar

There are mainly two types of radars used in ATC: primary (PSR) and secondary (SSR and Mode S). They measure range and azimuth, and in the case of SSR or Mode S, they also receive height from the aircraft barometer.

PSR and SSR can suffer from the presence of false alarms. Although some kinds of signal processing methods (such as CFAR detectors) try to alleviate these problems, false alarms can appear with different false alarm rates in different areas (due to potentially time changing effects such as weather, sea or terrain conditions).

A usual error model for the Mode-S and conventional secondary radar is described next. k-th range-azimuth measurement (R_k, θ_k) include the terms in (1):

$$\begin{aligned} R_k &= (1+K)R_{id}(k) + \Delta R + \Delta R_j + n_R(k) \\ \theta_k &= \theta_{id}(k) + \Delta\theta + n_\theta(k) \end{aligned} \tag{1}$$

where:

- ($R_{id}(k)$, $\theta_{id}(k)$) are the ideal target position for the k-th measurement, expressed in local polar coordinates, range and azimuth.
- ΔR: radar range bias, originated by drifts in the radar chain.
- K is the gain of range bias, mainly related to propagation.
- ΔR_j: transponder induced bias of j-th aircraft, is a target related bias term.
- $\Delta\theta$ is the azimuth bias, mainly due to drifts in the rotation servomechanism.
- ($n_R(k)$,$n_\theta(k)$) are measurement noise errors, mainly from thermal noise, quantification error, clock jitter or transponder reply jitter.

Primary radars share the same model except the lack of ΔR_j term.

2.2 ADS-B

ADS-B is based on the broadcasting of aircraft navigation information to ground through a data-link [5], using a unique identifier, known as ICAO address. Due to its cooperative nature, it does not suffer from the generation of false alarms, but sometimes outliers can appear due to coding/decoding issues.

With precise navigation systems as the ones being currently deployed in modern aircraft (WAAS, EGNOS or Galileo), ADS-B measurements suffer mainly from a time-stamping error, different for each aircraft. Older aircraft with less advanced navigation systems may also suffer other errors.

Therefore, the k-th position measurement (x_k, y_k), obtained using the stereographic projection over latitude, longitude and height measurements, may be modelled as:

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{id}(k) \\ y_{id}(k) \end{bmatrix} + H_{ADS} \begin{bmatrix} \Delta X_j \\ \Delta Y_j \\ \Delta t_j \end{bmatrix} + \begin{bmatrix} n_x(k) \\ n_y(k) \end{bmatrix}; \quad H_{ADS} = \begin{bmatrix} 1 & 0 & -V_X \\ 0 & 1 & -V_Y \end{bmatrix} \tag{2}$$

where:

- $x_{id}(k)$, $y_{id}(k)$ is the ideal target position for the k-th measurement.
- (V_X, V_Y) is the velocity vector of the target.
- $(\Delta X_j, \Delta Y_j)$: is a position offset vector for j-th aircraft.
- Δt_j: is the time offset for j-th aircraft.
- $n_x(k)$, $n_y(k)$ are measurement noise errors. Their variance is dependent of the navigation system quality, which is coded in the ADS-B message.

2.3 Wide Area Multilateration Error Models

Wide area multilateration measurement performs Time Difference Of Arrival (TDOA) estimation to calculate target position, based on the emission by the aircraft of random signals (secondary transponder squitters, responses to nearby radars, …) and its reception by a ground-based station network.

This system can suffer from the appearance of false alarms, due to presence of multipath leading to potential duplicated data as splits, but this is a problem much less frequent than with radar sensors (specially PSR), and a non-malfunctioning WAM sensor must have a very small false alarm rate or outlier rate.

The error of multilateration is a function of several variables, including:

- The geometry of the receiving stations and transmitter.
- The timing accuracy of the receiving stations. All stations will have a local clock with random errors (jitter) and slow changing errors (drifts).
- The accuracy of the synchronization of the receiving sites. The effect on position estimation due to drift errors must be approximately cancelled through accurate synchronization.

It should be noted that the multilateration system has internal calibration means, as without them no position estimation would be possible. This is a subject still under active research, a description of the main error terms may be found in [6].

We assume the position bias was corrected by the internal calibration system, while there can be a time offset with respect to fusion system reference clock. Then, the k-th position measurement (x_k, y_k), obtained using the stereographic projection over latitude, longitude and height measurements, may be modeled as:

$$\begin{aligned} x_k &= x_{id}(k) - V_X \Delta t + n_x(k) \\ y_k &= y_{id}(k) - V_Y \Delta t + n_y(k) \end{aligned} \tag{3}$$

where:

- $x_{id}(k)$, $y_{id}(k)$ is the ideal target position for the k-th measurement.
- (V_X, V_Y) is the velocity vector of the target.
- Δt: is the time bias, equal for all aircraft and cells. It is a sensor bias.
- $(n_x(k), n_y(k))$ are the white noise components in stereographic plane. Its covariance is usually provided with the WAM measurement.

3 Surveillance Sensors Assessment and Adaption

The proposed Multisensor Data fusion is depicted in Fig 1, where the usual Multisensor Multitarget Tracker (MMTT) has attached a parallel on-line Surveillance Sensor Assessment & Adaptation procedure. This procedure derives MMTT adaptation data from all sensor measurements and from Association data and multisensor/monosensor tracks.

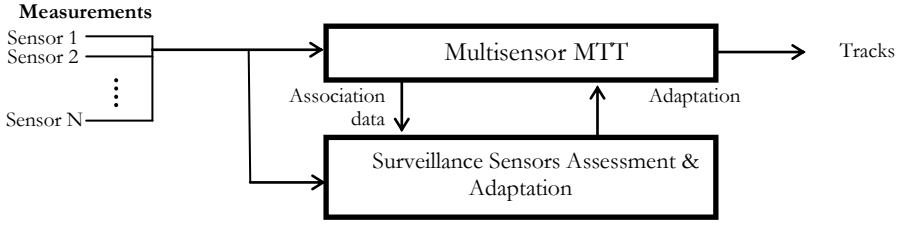


Fig. 1. Adaptive Multisensor Data Fusion Proposal

Surveillance Sensor Assesment and Adaptation is based on the observation of a set of data providing indirect information on not-modeled errors or situations, and in the update of sensor related models and algoritms within MMTT.

There are three different kinds of assessments:

- Sensor oriented, using all available data for a given sensor (with index i).
- Target oriented, using data from a given target. Those are specially important for ADS-B. We will use subindex t to indicate given aircraft.
- Grid oriented, segmenting data from a sensor in a 3D grid, to take into account variations of behavior depending on position. Radar grids are defined in polar coordinates while WAM grid is defined in stereographic coordinates. We will indicate given cell with index j.

Sensor Detection State Analysis is in charge of detecting anomalous detection or high outliers rate areas/sensors. The analysis is based on monosensor association/tracking.

To do so it performs a set of analysis:

1) Sensor/Grid estimation of track initiation rate (P_1).
2) Sensor/Grid estimation of track deletion rate (P_2).
3) Sensor/Grid estimation of track life duration statistics of deleted tracks (P_3).
4) Sensor/Grid estimation of non-associated measures rate (P_4).
5) Sensor/Grid estimation of ambiguous data association rate (P_5).
6) Target oriented number of tracks for ADS-B ICAO address (P_6).

Analysis 1) to 4) are related to high false alarm rates. 5) is related to presence of crossing targets or false alarms and splits. In an MMTT these situations are alleviated by hardening track initiation/confirmation requirements (by tuning those algorithms parameters), and reducing association gates around predicted tracks to reduce the probability of association of a false alarm to a track. 6) is an special case, accounting for the case of an unstable target measurement process, indication of a potentially malfunctioning ADS-B equipment.

Variance Assessment is another interest test. Performing differences of consecutive measures from a given sensor and target we may obtain an observation related to sensor noise, assuming constant velocity dynamics. From a collection of such measures sensor based variances may be derived. Note ADS-B and WAM provide covariances of measure, so a consistency analysis between sensor provided variances and observed variances may be performed.

In this case we also have a set of analysis:

1) Sensor/Grid estimation of covariance for radar (S_1).
2) Sensor/Grid Consistency between sensor provided covariance and measured covariance for WAM (S_2).
3) Target oriented consistency analysis between measured covariance and ADS-B provided covariance (S_3).

Each of previous analysis can provide better-adapted covariance data (enabling improved statistical data fusion) or serve as the basis for the detection of malfunctioning sensors.

Bias Assesment, in parallel to bias Estimation and correction is a key element for alignment of sensor data. It is based on the analysis of the mismatch between different sensor measures, taking into account the previous characterization of this mismatch. It obtains the lists of sensor and target biases (to be denoted as b_0 and b_1).

It estimates all previous bias terms in a sensor/target-oriented manner. In addition, after performing tracking, the offset between bias-corrected measures and predicted tracks is averaged in a grid, in order to localize areas with additional bias terms not corrected, potentially due to malfunctioning sensors. Let's call these values b_2.

Monosensor/Multisensor Compatibility Assessment provides additional integrity to our proposal. Multisensor tracks are based on data from all sensors, while monosensor tracks have only one sensor measures. A statistical compatibility assessment of all monosensor tracks and multisensor tracks, taking into account sensor coverages may be used to detect uncorrected bias terms (specially for target related biases), or, by analyzing the presence of missed detections, malfunctioning sensor.

The main observations here are:

1) Sensor/Grid/Target Monosensor/Multisensor track offset compatibility assessments (C_1).
2) Sensor/Grid/Target missed measure analysis for radar (C_2).

3.1 Adaptation Expert System Design

An expert system has been designed to take decisions that make the whole system perform better when some sensor or target's measures stop behaving as expected. The adaptation decisions the expert system is able to take for each sensor, grid or target include changing the track initialization method, changing the size of the association windows, updating the measurement models used for filtering or completely removing measurements under certain conditions.

The knowledge used in the system is structured as a set of rules. These rules are used to inference the conclusions (in this work, conclusions are the need to take any of the adaptation decisions described above) using the sensor detection state, variance assessment, bias assessment and monosensor/multisensor compatibility assessment data.

Out system has more than 50 different rules, most of which can be inferred from previous discussion. Due to lack of space here we are just providing some of them:

- For each sensor i, and grid cell j (all actions are for sensor-cell i,j).

if $P_1(i,j)>u_{1,1}$ **and** $P_3(i,j)<u_{3,1}$ **and** $P_4(i,j)>u_{4,1}$ **then** harden initialization
if $P_1(i,j)>u_{1,2}$ **and** $P_2(i,j)>u_{2,2}$ **and** $P_3(i,j)<u_{3,2}$ **and** $P_4(i,j)>u_{4,2}$ **then** disable initialization
if $S_1(i,j)>nominal\_S_1(i)$ **or** $C_1(i,j)>u_{9,1}$ **then** increase radar measure covariance matrix
if $S_2(i,j)$=inconsistent **or** $C_1(i,j)>u_{9,2}$ **then** increase WAM measure covariance matrix
if $P_1(i,j)>u_{1,3}$ **or** $P_2(i,j)>u_{2,3}$ **or** $P_3(i,j)<u_{3,3}$ **or** $P_4(i,j)>u_{4,3}$ **or** $P_5(i,j)>u_{5,3}$ **or** $S_1(i,j)>>nominal\_S_1(i,j)$ **or** $S_2(i,j)$= very inconsistent **or** $b_0(i,j)>u_6$ **or** $b_2(i,j)>u_8$ **or** $C_1(i,j)>u_{9,3}$ **then** disable use of measures for association and tracking

- For each sensor i (all actions are performed for measures from i-th sensor).

if $P_1(i)>w_{1,1}$ **and** $P_3(i)<w_{3,1}$ **and** $P_4(i)>w_{4,1}$ **then** harden initialization
if $P_1(i)>w_{1,2}$ **and** $P_2(i)>w_{2,2}$ **and** $P_3(i)<w_{3,2}$ **and** $P_4(i)>w_{4,2}$ **then** disable initialization
if $S_1(i)>nominal\_S_1(i)$**or** $C_1(i)>w_{9,1}$ **then** increase radar measure covariance matrix
if $S_2(i)$=inconsistent **or** $C_1(i)>w_{9,2}$ **then** increase WAM measure covariance matrix
if $P_1(i)>w_{1,3}$ **or** $P_2(i)>w_{2,3}$ **or** $P_3(i)<w_{3,3}$ **or** $P_4(i)>w_{4,3}$ **or** $P_5(i)>w_{5,3}$ **or** $S_1(i)>>nominal\_S_1(i)$ **or** $S_2(i)$= very inconsistent **or** $b_0(i)>w_6$ **or** $b_2(i)>w_8$ **or** $C_1(i)>u_{9,3}$ **then** disable use of measures for association and tracking

- For each target t (all actions performed for measures from t-th target)

if $P_6(i,t)>z_6$ **then** disable initialization from ADS-B
if $S_3(t)<z_3$ **or** $C_1(t)>z_{9,1}$ **then** increase ADS-B measure covariance matrix
if $b_1(t)<z_7$ **or** $C_1(t)>z_{9,2}$ **then** disable use of measures for association and tracking

Some other rules are aimed to restore the default behaviour of the MMTT when the conditions of the sensors return back to the expected ones. For instance, when measured covariances return to nominal values the increased covariance matrix is not used anymore.

4 Simulation Results

We have conducted a set of simulations and worked with real data in order to be able to tune the different thresholds in the expert system rules, to increase overall system integrity. Due to lack of space and confidentiality requirements over real data, only one simulated scenario will be described next. In this simple scenario we have five radars with overlapping coverage, two of them are PSR, and the others are Mode S. The simulated bias does not follow the described model for one of the PSR, and therefore, after bias correction, there is a remaining uncorrected bias leading to a mismatch of around 200 meters in radial coordinate and 0.2º in azimuth for a certain area of the radar coverage. Due to that, $b_2(i,j)$ and $C_1(i,j)$ exceed their corresponding thresholds and therefore, for this cell, which leads to precluding the use of this sensor measure to perform tracking. The results, interpolated for different times, for a given trajectory, are summarized in next figure.

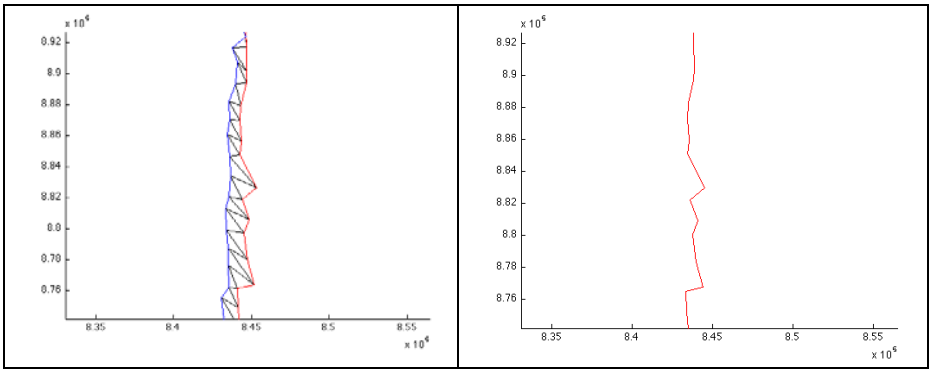


Fig. 2. Adaptive Multisensor Data Fusion Proposal

In the left part of the figure the monosensor tracks from two radars feeding the track, in red and blue, and the multisensor track in black (with extremely problematic velocity vector). Meanwhile, in the right, after removal of the problematic sensor, only one of the monosensor track is available which becomes equivalent to the multisensor track.

5 Conclusion

In this paper we are proposing the use of a set of figures of merit, coupled with an expert system, to provide a Multisensor Multitarget Tracker with sensor context adaptation capabilities. Such tests are performed using the information provided by the sensors and the MMTT, and the output of the tests is defined by a rule based expert system which takes decisions and makes changes in the MMTT by changing parameters or disabling sensors in order to obtain better results.

Some Air Traffic Control such as ARTAS [7] or SACTA [8] surveillance systems already include simple adaptation mechanisms. However, such mechanisms are

usually tightly bounded to the MMTT itself, which makes them more difficult to maintain or improve. This work aims to extract these mechanisms and place them in a separate procedure, which allows expressing the adaptation knowledge in a more formal way. Also, having a Surveillance Assessment & Adaptation process eases future enhancements of the context adaptation mechanisms, as all the rules are centralized.

This is a new research thread that will be further developed by our group. Future enhancements include automatic optimal threshold search and applying machine learning techniques to the context adaptation process.

Acknowledgments. This work was funded by contract Spanish Research Ministry, under contracts CICYT TSI2005-07344, CICYT TEC2005-07186.

References

1. Rong Li, X., Bar-Shalom, Y.: Design of an Interacting Multiple Model Algorithm for Air Traffic Control Tracking. IEEE Transactions on Control System Technology 1(3) (Septemper 1993)
2. Rafati, A., Moshiri, B., Rezaei, J.: A new algorithm for general asynchronous sensor bias estimation in multisensor-multitarget systems. In: Proceeding of the 10th International Conference on Information Fusion, Quebec, Canada (2007)
3. Lin, X., Bar-Shalom, Y., Kirubarajan, T.: Multisensor multitarget bias estimation for general asynchronous sensors. IEEE Transactions on Aerospace and Electronic Systems 41, 899–921 (2005)
4. Lin, X., Bar-shalom, Y., Kirubarajan, T.: Multisensor-multitarget bias estimation for general asynchronous sensors. IEEE Transactions on Aerospace and Electronic Systems 41(3), 899–921 (2005)
5. ADS-B/TIS-B Functional Architecture, Draft version 0.7. In: EUROCONTROL (July 30, 2003)
6. Leeson, M.J.: Error Analysis for a wide area multilateration system. QinetiQ/ C&IS/ ADC/ 520896/7/19
7. Hoogendoorn, R.A., Neven, W.H.L.: ARTAS: Multisensor tracking in an ATC environment. NLR-TP-97657
8. `http://www.aena.es`

Dynamic Channel Model LMS Updating for RSS-Based Localization

Paula Tarrío, Ana M. Bernardos, Xian Wang, and José R. Casar

Data Processing and Simulation Group, Universidad Politécnica de Madrid, Madrid, Spain
{paula,abernardos,wang.xian,jramon}@grpss.ssr.upm.es

Abstract. Received signal strength-based localization systems usually rely on a calibration process that aims at characterizing the propagation channel. However, due to the changing environmental dynamics, the behavior of the channel may change after some time, thus, recalibration processes are necessary to maintain the positioning accuracy. This paper proposes a dynamic calibration method to initially calibrate and subsequently update the parameters of the propagation channel model using a Least Mean Squares approach. The method assumes that each anchor node in the localization infrastructure is characterized by its own propagation channel model. In practice, a set of sniffers is used to collect RSS samples, which will be used to automatically calibrate each channel model by iteratively minimizing the positioning error. The proposed method is validated through numerical simulation, showing that the positioning error of the mobile nodes is effectively reduced. Furthermore, the method has a very low computational cost; therefore it can be used in real-time operation for wireless resource-constrained nodes.

Keywords: Indoor localization, RSS, calibration, channel model, LMS.

1 Introduction

Many context-aware applications rely on the knowledge of the position of the user, and of the surrounding objects, to provide him with useful and personalized information and services. In indoor environments, where GPS cannot be used, several technologies have been proposed to calculate the position of a person or object, such as ultrasounds, artificial vision or infrared. However, due to the widespread use of wireless devices, radio-frequency localization techniques [1] and, in particular, those based on the measurement of the received signal strength (RSS), have become very popular and easy to deploy.

Either map-based or channel model based techniques can be used to locate a node from a set of RSS measurements. Channel model based techniques use a propagation channel model to establish a relation between the RSS and the distance between two nodes; then, a triangulation or positioning algorithm is used to calculate the position of a node from a set of distances to some anchor nodes with known positions. Map-based or fingerprinting techniques create a radio map of the environment by

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 127–135, 2011.

gathering, for each anchor node, a set of RSS measurements in different test points. When an unknown node needs to be localized, its RSS measurements are matched against the ones stored in the map, in order to find the closest correspondence. Both approaches require an initial calibration phase to obtain an appropriate fingerprint or channel model, valid for the specific deployment area.

The localization accuracy will depend on how accurately the propagation channel is characterized. The temporal variations of the propagation medium, originated by unstable environmental conditions (such as humidity), space reorganization (e.g. furniture movement or open-closed doors) and people's movement (temporal flow, human clusters around the mobile target, etc.), may therefore affect the localization accuracy. For example, [2] analyzes how the average position accuracy of a fingerprint-based system (offering 2.13m. of accuracy in standard conditions – no-blocking people, close-all-doors and 40% humidity level) is deteriorated in a 43.7% when the humidity level increases until 70%, in a 236.6% if the configuration changes to all-open-doors, and in a 85.9% when people clusters are present.

Due to the dynamic behavior of the propagation channel, the initial calibration may not be accurate enough after some time, so there is a need to repeat the calibration process in order to maintain the localization accuracy. In this paper, we present an automatic recalibration strategy for channel model-based localization systems, which is an enhanced version of our previous work [3]. We assume that the propagation channel adjusts to a theoretical lognormal model, with different parameters characterizing each of the anchor nodes. The proposed technique uses a set of reference points (with known positions) where RSS measurements from the anchor nodes are collected (either by a set of sniffers deployed at the reference points or by a mobile user that is detected at these particular points). These measurements are then integrated in a Least Mean Squares (LMS) algorithm that finds, iteratively, the values of the channel parameters that minimize the positioning error. The proposed technique can be implemented in real-time localization systems, as its computational and memory requirements are very low.

The structure of the paper is as follows. Section 2 reviews some previous proposals to automate calibration procedures for indoor localization systems. In Section 3, our localization scenario is fully described. Section 4 describes the proposed algorithm, which is tested in Section 5 through a number of numerical simulations. Finally, Section 6 concludes the work.

2 Related Work

RSS indoor localization systems usually rely on an off-line calibration phase, which aims at characterizing the electromagnetic environment by: 1) calibrating a theoretical propagation model with real RSS measurements or 2) building a RSS fingerprint (or radio map) of the localization area. Due to the changing environmental dynamics the reliability of the calibration does not last for a long time, thus, recalibration processes are necessary to maintain the positioning accuracy. Manual calibration and recalibration processes are costly and inefficient, so a line of research in indoor localization is devoted to propose solutions with zero or limited initial calibration and on-line recalibration.

Several fingerprinting localization approaches propose to update the radio map automatically without human interaction. This is usually achieved by using additional devices that listen to the transmitted signals. For example, Krishnan et al. [4] include sniffers in their RF deployment. When at least one sniffer observes a significant deviation on the RSS from any emitter, the radio map is recalculated using a spline interpolation technique on the sniffers data. Moraes and Nunes [5] also propose a sniffer-based technique to build a propagation map, in which each grid position is associated to a probability distribution. The map is rebuilt every T seconds or when significant variations in the RSS occur. A similar approach is followed by Yin et al. [6], who use a set of reference points (whose position is not needed) to measure the RSS from the anchor nodes of the deployment. These measurements and the ones collected from a mobile node are used in a multiple-regression based algorithm to update a linear relationship between the signal-strength values received by the reference points and those received by the client device.

With respect to channel modeling localization techniques, several methods have been proposed to calculate and update the channel model online. For example, Gwon and Jain [7] use inter-anchor RSS measurements to generate multiple linear functions (one for each pair of anchors) representing the relationship between RSS and distance. When the mobile node needs to be localized, it uses the mapping function corresponding to the first and second anchors with strongest RSS to convert the RSS into distance. A similar approach is followed by Barsocchi et al. [8]; they use the inter-anchor RSS measurements to calculate, adaptively, a RSS-distance model that, in this case, is logarithmic and includes, as parameters to update, a wall attenuation factor and the air attenuation factor (or path loss exponent). Lim et al. [9] take as input the on-line RSS measurements between anchors, and between a client and its neighboring anchors, to create a linear mapping between RSS and distance using the truncated singular value decomposition technique. The algorithm implicitly assumes a logarithmic path loss model and that the distance between a client and an anchor node is a linear combination of the RSS measurements between the client and all the anchor nodes. Our previous work [3] uses real-time RSS measurements from the anchor nodes obtained from a set of reference points to update the parameters of a logarithmic propagation model by using a LMS algorithm. We propose here an extension of this method that considers a different propagation model for each of the anchor nodes. In this way, the accuracy of the localization results is improved when the deployment area is such that different anchor nodes may be affected by different propagation conditions.

3 Localization Scenario

Our calibration scheme is targeted at dynamically adjusting the propagation channel models in model-based localization systems. This kind of systems are usually composed of N anchor nodes (e.g. WiFi or Bluetooth access points, or Zigbee motes) with fixed and known positions, and one or several mobile targets that need to be localized. The localization is based on using a channel model to compute each mobile-anchor node distance from the RSS measurements taken at the mobile device from the anchor nodes (or vice versa). The position of the target is then computed with a triangulation or positioning algorithm.

The most popular channel model for RSS-based localization is the lognormal model [10]:

$$P_{RX}(dBm) = A - 10\eta \log \frac{d}{d_0} + N(0,\sigma) \tag{1}$$

where P_{RX} is the received power, d is the distance between transmitter and receiver, A and η are the parameters of the channel model and N is a zero-mean Gaussian random variable with standard deviation σ. A depends on the antenna gains, the transmission power and the power loss for a reference distance d_0, and needs to be experimentally adjusted. The path loss exponent η has to be experimentally determined too. For example, in 2.4GHz IEEE 802.15.4 propagation, considering d_0 = 1 m, A may range between -50 and -85 dBm, while η may be between 1.9 and 3.5 [11].

In our case, we assume that each anchor node is characterized by different values of the parameters A and η of the lognormal channel model. Therefore, the distance between any point and anchor node i can be estimated from the received P_{RX} (in practice the *RSS*) using eq. 1 and given A_i and η_i. Then, from a set of, at least, three estimated distances to different anchor nodes, the target's position can be calculated. To this end, we use the hyperbolic positioning algorithm (detailed formulation is available in e.g. [10]), which estimates the position of the target according to the following expression:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \left(H^T H\right)^{-1} H^T \tilde{b} \tag{2}$$

where:

$$H = \begin{bmatrix} 2x_2 & 2y_2 \\ \vdots & \vdots \\ 2x_N & 2y_N \end{bmatrix}; \quad \tilde{b} = \begin{bmatrix} x_2^{\,2} + y_2^{\,2} - \tilde{d}_2^{\,2} + \tilde{d}_1^{\,2} \\ \vdots \\ x_N^{\,2} + y_N^{\,2} - \tilde{d}_N^{\,2} + \tilde{d}_1^{\,2} \end{bmatrix} \tag{3}$$

$$\tilde{d}_i = 10^{\frac{A_i - RSS_i}{10\eta_i}} \tag{4}$$

being (x_i, y_i) the known coordinates of anchor node i, $\tilde{d}_i$ the estimated distance from the target to anchor node i, and RSS_i, the received signal strength to/from anchor node i. The origin of coordinates is situated in the anchor node $i=1$ (x_1=0, y_1=0).

In practice, the parameters A_i and η_i need to be continually updated or calibrated, as slightly biased estimations of A and η may result in significant localization errors [10]. To do so automatically, we define a number of reference points at fixed geographic positions, where a wireless device will take RSS measurements from the anchor nodes, which will be used to update the propagation model, according to the algorithm described in next section. These reference points, may be related to waypoints or objects capable of generating 'measurement events' (e.g. doors that detect users), or deployed as part of the communications infrastructure (i.e. an anchor node could serve as reference point).

Of course, this approach has many practical implementation details that are not directly addressed in this paper. For example, the number of reference points needs to be minimized, and the physical distribution of the reference points needs to be

flexible, as it is not always easy to place new elements in daily-living environments. Additionally, reference points should be easily maintainable and admit dynamic reconfiguration. Assuming that a suitable deployment is feasible (as it is), the optimization algorithm used to calibrate the system is described in the next Section.

4 Proposed Adaptive Calibration Algorithm

The Least Mean Square algorithm is a kind of stochastic gradient algorithm, based on approximating the true gradient of the mean-square error of a function by its instantaneous estimate. The LMS algorithm is a simple and computationally efficient technique used to find the values of the parameters of a function that fit to a set of reference values.

In this case, we propose to use it in an adaptive filter to minimize the localization error (eq. 5) by recursively adapting the parameters A_i and η_i of the lognormal propagation models. A block diagram of the proposed scheme is shown in Fig. 1.

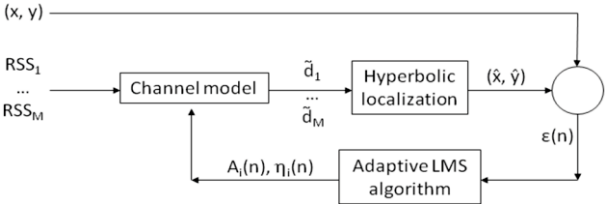


Fig. 1. Block diagram of the adaptive LMS algorithm

At each new iteration (n), the filter takes as input $M \leq N$ RSS measurements taken between a certain calibration/reference point and M anchor nodes. These measurements are used to estimate the distances $\tilde{d}_i$ from this reference point to the set of anchor nodes, using eq. 4 and the values of A_i and η_i calculated at the previous iteration (n-1). Then, the hyperbolic positioning algorithm (eq. 2) is used to estimate from these distances the position of the reference point $(\hat{x}, \hat{y})$, which is compared with its known real position (x, y) to evaluate the error:

$$\varepsilon(n) = \sqrt{\left(x(n) - \hat{x}(n)\right)^2 + \left(y(n) - \hat{y}(n)\right)^2} \tag{5}$$

This error serves as input to the LMS algorithm, which finds the parameters A_i and η_i at the current iteration n that minimizes this error. According to the LMS technique, the optimum values of these parameters can be calculated as:

$$A_i(n) = A_i(n-1) - \frac{1}{2}\mu_{Ai}\frac{\partial E[\varepsilon^2(n)]}{\partial A_i} \cong A_i(n-1) - \mu_{Ai}\varepsilon(n)\frac{\partial \varepsilon(n)}{\partial A_i} \tag{6}$$

$$\eta_i(n) \cong \eta_i(n-1) - \mu_{\eta i}\varepsilon(n)\frac{\partial \varepsilon(n)}{\partial \eta_i}$$

where μs are the filter step sizes, which control the speed and stability of convergence, and the partial derivatives can be obtained from eqs. 2-5 after some calculations:

$$\frac{\partial \varepsilon(n)}{\partial A_i} = \frac{1}{\varepsilon(n)} \cdot \frac{16 \cdot \ln 10}{\det \cdot 10 \cdot \eta_i (n-1)} \cdot \left[C_i \cdot \left(x(n) - \hat{x}(n) \right) + D_i \cdot \left(y(n) - \hat{y}(n) \right) \right] \cdot \tilde{d}_i^{\,2} \qquad i = 2..N$$

$$\frac{\partial \varepsilon(n)}{\partial A_1} = -\frac{1}{\varepsilon(n)} \cdot \frac{16 \cdot \ln 10}{\det \cdot 10 \cdot \eta_1 (n-1)} \cdot \left[\sum_{i=2}^{N} C_i \cdot \left(x(n) - \hat{x}(n) \right) + \sum_{i=2}^{N} D_i \cdot \left(y(n) - \hat{y}(n) \right) \right] \cdot \tilde{d}_1^{\,2}$$

$$\frac{\partial \varepsilon(n)}{\partial \eta_i} = -\frac{1}{\varepsilon(n)} \cdot \frac{16 \cdot \ln 10}{\det \cdot \eta_i (n-1)} \cdot \left[C_i \cdot \left(x(n) - \hat{x}(n) \right) + D_i \cdot \left(y(n) - \hat{y}(n) \right) \right] \cdot \tilde{d}_i^{\,2} \log \tilde{d}_i \qquad i = 2..N \tag{7}$$

$$\frac{\partial \varepsilon(n)}{\partial \eta_1} = \frac{1}{\varepsilon(n)} \cdot \frac{16 \cdot \ln 10}{\det \cdot \eta_1 (n-1)} \cdot \left[\sum_{i=2}^{N} C_i \cdot \left(x(n) - \hat{x}(n) \right) + \sum_{i=2}^{N} D_i \cdot \left(y(n) - \hat{y}(n) \right) \right] \cdot \tilde{d}_1^{\,2} \log \tilde{d}_1$$

where

$$\tilde{d}_i = 10^{\frac{A_i(n-1) - RSS_i}{10 \cdot \eta_i (n-1)}}$$

$$C_i = x_i \cdot \left(y_2^2 + \ldots + y_N^2 \right) - y_i \cdot \left(x_2 \cdot y_2 + \ldots + x_N \cdot y_N \right), \qquad D_i = y_i \cdot \left(x_2^2 + \ldots + x_N^2 \right) - x_i \cdot \left(x_2 \cdot y_2 + \ldots + x_N \cdot y_N \right)$$

$$\det = \left(4 \cdot x_2^2 + \ldots + 4 \cdot x_N^2 \right) \cdot \left(4 \cdot y_2^2 + \ldots + 4 \cdot y_N^2 \right) - \left(4 \cdot x_2 \cdot y_2 + \ldots + 4 \cdot x_N \cdot y_N \right)^2$$

As it can be noticed, the proposed iterative technique just handles data of the previous temporal instant and is simple in its formulation, thus, it has minimum computational and memory needs. Therefore, it can be integrated in real-time localization systems without requiring significant resources or introducing serious operational delays.

5 Performance Evaluation

In order to evaluate the performance of the proposed method, Matlab was used to simulate a wireless deployment in a noisy environment. The simulation scenario was composed of 8 anchor nodes and 20 reference points, as shown in Fig. 2.

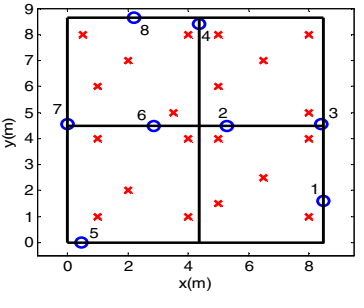


Fig. 2. Scenario for simulation: 4 rooms, 8 anchor nodes (o) and 20 reference points (x)

The positions of the reference points were chosen to have enough spatial diversity in each room with a moderate number of nodes (5 per room, located near the corners

and the center). For each reference node, we generate 200 data arrays, each one containing RSS measurements from the 8 anchor nodes. These RSS measurements are simulated using eq. 1, with $P_{TX\text{-}sim}$ = 0 dBm, $d_{0\text{-}sim}$ = 1 m and σ_{sim} = 1 dB. The channel parameters were set to $A_{sim\text{-}i}$ = -60 dB and $\eta_{sim\text{-}i}$ = 2.3 for the first four anchor nodes (i=1..4) and to $A_{sim\text{-}i}$ = -65 dB and $\eta_{sim\text{-}i}$ = 2.6 for the other four (i=5..8).

The algorithm is initialized with a set of initial values for the parameters of the channel models of the different anchor nodes (A_{0i} and η_{0i}). Then, in the first iteration, one of the reference points provides a set of RSS measurements from the 8 anchor nodes. The initial channel model parameters are used to convert these measurements into distances (using eq. 1), which are introduced into the hyperbolic localization algorithm (eq. 2) to obtain a first position estimation of the reference point. Then, equations 6-7 are used to update the values of A_i and η_i that minimize the position error. Those values are to be used by the localization algorithm in the next iteration.

Fig. 3 shows the evolution of the position error (Euclidean distance between the real position and the estimated position) for a set of 100 mobile nodes randomly distributed in the simulation area. In this case, the initial values of the channel model parameters were set to A_0 = -65 dB and η_0 = 2.4 for the first 4 anchors, and to A_0 = -60 dB and η_0 = 2.7 for the other four. It can be seen that the algorithm converges in this case after approximately 25 samples (i.e., 25 measurement events, with 8 RSS measurements at each event: one for each anchor node). When the difference between the initial values of the channel model parameters and their "true" value is higher, the convergence is slower. For example, when if the initial values of the channel model parameters are set to A_0 = -70 dB and η_0 = 2.4 for the first 4 anchors, and to A_0 = -55 dB and η_0 = 2.7 for the other four, approximately 100 samples are needed to calibrate the model (Fig. 3b).

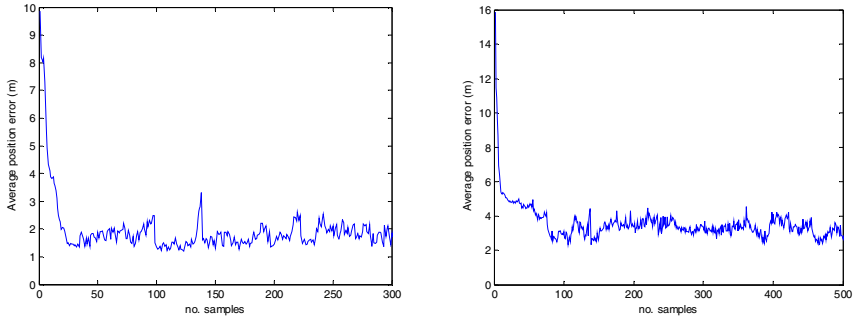


Fig. 3. Evolution of the positioning error using a LMS filter with μ_A = 0.8, μ_η = 0.01. The initial values of the model parameters were a) A_i = -65 dB and η_i = 2.4 for i = 1..4, and A_i = -60 dB and η_i = 2.7 for i = 5..8, b) A_i = -70 dB and η_i = 2.4 for i = 1..4, and A_i = -55 dB and η_i = 2.7 for i = 5..8.

As comparison, we have also evaluated the convergence of the real-time calibration algorithm proposed in [3], which also uses a LMS technique to minimize the error, but assumes a unique and isotropic channel model for the entire environment. Fig. 4 shows the evolution of the positioning error under the same simulation environment as in Fig. 3a. As it can be seen, the new method provides better results: the convergence is quicker and the final positioning error is lower.

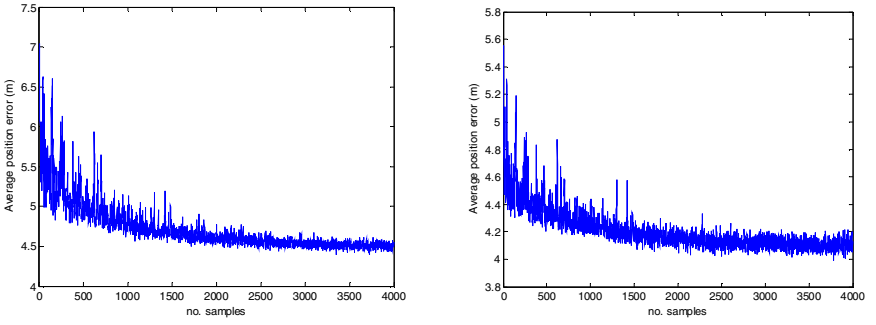


Fig. 4. Evolution of the positioning error using the LMS filter in [3] with $\mu_A = 0.8$, $\mu_\eta = 0.01$. The initial values of the model parameters were a) A = -65 dB and $\eta = 2.4$, b) A = -60 dB and $\eta = 2.7$.

A significant factor for the LMS algorithm are the filter step sizes (μs), which control the pace to convergence but also the stability of the estimation. For our experiments, the step sizes have been empirically chosen among those that where offering a reasonable convergence time (in terms of needed number of RSS tuples) while providing a reasonably stable convergence value for A_i and η_i. Although there are other values which may be effectively used, Fig. 3 shows how the convergence works for the distance error when the filter step sizes have been set to $\mu_A = 0.8$, $\mu_\eta = 0.01$. As shown in Fig. 5, higher values of μ may accelerate the convergence (fewer samples are needed to reach the possible minimum error in distance), but may also provide a less stable convergence.

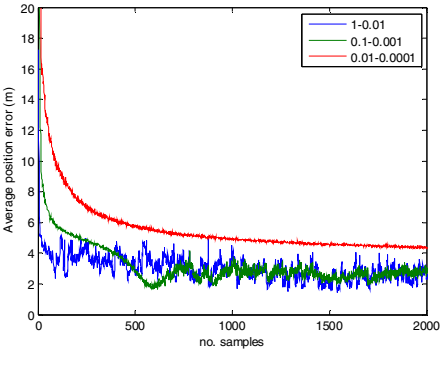


Fig. 5. Effects of the value of the LMS filter coefficients $<\mu_A, \mu_\eta>$ on the mean error evolution (in m.). LMS initialization: A_i = -70 dB and $\eta_i = 2.4$ for $i = 1..4$, and A_i = -55 dB and $\eta_i = 2.7$ for $i = 5..8$.

6 Conclusions

This paper describes a strategy to automatically adapt the parameters of the lognormal channel models in order to minimize localization errors when using a hyperbolic

localization technique. Preliminary numerical results show that the strategy achieves good localization results after a short convergence time. Furthermore, its computational and memory requirements are very low. Therefore, it is a promising technique to use in resource-constrained devices to automate the calibration procedure.

From a practical viewpoint, we have deployed a testbed with MicaZ devices and we are starting to carry out some experimental test to assess the performance of the algorithm in a real situation. Further work is focused on demonstrating the stability and feasibility of the proposal in real time operation; in particular, it is necessary to study how to set the LMS step sizes effectively, and to analyze the relationship between the number of reference points, their geometrical distribution and the LMS algorithm accuracy.

Acknowledgements. This work has been supported by the Government of Madrid under grant S-2009/TIC-1485 and by the Spanish Ministry of Science and Innovation under grant TIN2008-06742-C02-01.

References

1. Gezici, S.: A survey on wireless position estimation. Wireless Personal Communications 44(3), 263–282 (2008)
2. Chen, Y.C., Chiang, J.R., Chu, H.H., Huang, P., Tsui, A.W.: Sensor-assisted WiFi Indoor Location System for Adapting to Environmental Dynamics. In: 8th ACM Int. Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 118–125 (2005)
3. Bernardos, A., Casar, J.R., Tarrío, P.: Real time calibration for RSS indoor positioning systems. In: Int. Conf. on Indoor Positioning and Indoor Navigation (2010)
4. Krishnan, P., Krishnakumar, A.S., Ju, W.-H., Mallows, C., Ganu, S.: A system for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks. In: IEEE INFOCOM, vol. 2, pp. 1001–1011 (2004)
5. Moraes, L.F.M., Nunes, B.A.A.: Calibration-Free WLAN Location System Based on Dynamic Mapping of Signal Strength. In: 4th ACM Int. Workshop on Mobility Management and Wireless Access, pp. 92–99 (2006)
6. Yin, J., Yang, Q., Ni, L.: Adaptive Temporal Radio Maps for Indoor Location Estimation. In: 3rd IEEE Int. Conf. on Pervasive Computing and Communications (2005)
7. Gwon, Y., Jain, R.: Error Characteristics and Calibration-free Techniques for Wireless LAN-based Location Estimation. In: 2nd Int. Workshop on Mobility Management & Wireless Access Protocols, pp. 2–9. ACM Press, New York (2004)
8. Barsocchi, P., Lenzi, S., Chessa, S., Giunta, G.: A Novel Approach to Indoor RSSI Localization by Automatic Calibration of the Wireless Propagation Model. In: 69th Conf. on Vehicular Technology, pp. 1–5 (2009)
9. Lim, H., Kung, L.-C., Hou, J.C., Luo, H.: Zero-configuration, robust indoor localization: Theory and Experimentation. In: IEEE INFOCOM, pp. 1–12 (2006)
10. Tarrío, P., Bernardos, A.M., Casar, J.R.: An RSS Localization Method based on Parametric Channel Models. In: Int. Conf. on Sensor Technologies and Applications, pp. 265–270 (2007)
11. Tarrío, P., Martín, H., Bernardos, A.M.: Enhancing the Performance of Propagation Model-Based Positioning Algorithms. In: 3rd Int. Workshop on User-Centric Technologies and Applications, pp. 123–132 (2009)

Improving the Accuracy of Action Classification Using View-Dependent Context Information

Rodrigo Cilla, Miguel A. Patricio, Antonio Berlanga, and Jos M. Molina

Computer Science Department, Universidad Carlos III de Madrid
Avda. de la Universidad Carlos III, 22. 28270 Colmenarejo. Madrid, Spain
{rcilla,mpatrici}@inf.uc3m.es,{berlanga,molina}@ia.uc3m.es

Abstract. This paper presents a human action recognition system that decomposes the task in two subtasks. First, a view-independent classifier, shared between the multiple views to analyze, is applied to obtain an initial guess of the posterior distribution of the performed action. Then, this posterior distribution is combined with view based knowledge to improve the action classification. This allows to reuse the view-independent component when a new view has to be analyzed, needing to only specify the view dependent knowledge. An example of the application of the system into an smart home domain is discussed.

1 Introduction

The recognition of human actions from video has been a very active research field during the last two decades. Video Surveillance, Multimedia indexing and retrieval or Ambient Assisted Living are some of the applications that have been benefited from the advances made during this period [12].

One of the most recent research issues on human action recognition is the usage of viewpoint independent action representations. The objective is to obtain a representation of the action robust to changes in the viewpoint of the camera that grabs the processed images. The problem has been studied from different perspectives. Some approaches [13,10,6,16] rely on imposing geometrical constraints on 3D body limbs configurations. Its use is limited to situations where an accurate body part tracker is available. Others try to study 3D visual hull evolutions of the human body [22,11,20]. Some proposals associate 2D silhouettes extracted from a single camera with their corresponding 3D visual hulls [8,21]. View-independent action classification can be achieved with non-linear classifiers, [9,19], achieving a promising accuracy. Recently, view-invariant actions representations have been learned in low dimensional manifolds [14,7].

Obtaining a true view invariant representation of the action would allow to share action models between different scenarios, reducing the cost of creating new action recognition systems. The view-invariant action primitives then would be used from a library, only requiring to define scenario dependent action semantics.

By the other hand, the likelihood observing a given action is very dependent on the features of the scene being analyzed. The action *sit* is going to be more

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 136–143, 2011.

likely to happen in a place where there is a chair, or the action *walk* in a clear area of the scene. The scene context information can help us to reason about what actions are more likely to be performed by the observed human. Combining view-invariant action classifiers with context information might then be a way to improve the final system accuracy.

Context information has been used to improve the performance of different computer vision tasks. Gomez-Romero et al. [5] propose the usage of context information to improve the performance of object tracking systems. Robertson and Reid [15] proposed a probabilistic discriminative framework to combine different attributes of human action recognition, improving optical flow action classification using position and speed knowledge. In a related work, Wu and Aghajan [23] use the actions of the user as the context information to discover the objects in the environment.

This paper presents an action recognition system combining view-invariant classifiers with context information. A viewpoint independent classifier (VIC) is trained using samples of actions obtained from different cameras. For a given view, we learn a context probabilistic model (CPM) relating the positions of the human bounding boxes in the view and the actions. To decide what is the action happening on a new observation the output of the VIC is averaged by the CPM, making the final result.

Paper is organized as follows. In section 2 we present the general structure of the system; in section 3 the view-independent classifier is introduced; in section 4 the view-dependent model used as context is presented; in section 5 we show how the system can be used in a smart home. Finally, in section 6, the conclusions and future research lines of this work are presented.

2 System Overview

The architecture of the proposed system can be observed on figure 1. At a given instant t, a image $I_t(x, y)$ of the scene is grabbed from a fixed viewpoint. A foreground mask $F_t(x, y)$ containing the humans of interest is extracted using background subtraction [17]. The bounding boxes of the objects of interest, i.e., the people in the view, are tracked over time, with a method such the proposed in [4]. For simplicity we restrict to the case of just one person being observed. His bounding box is represented by a 4d vector $b_t = (x, y, w, h)$, where the first two components correspond to the centroid and the last two to the size. The feature extractor (FE) extracts the attribute vector f_t and feds it into the view invariant classifier (VIC) to obtain a preliminary posterior distribution of the action α_t performed, $p(\alpha_t \mid f_t)$, $\alpha_t \in A = \{a_1, \ldots, a_N\}$. This distribution is averaged by the context probabilistic model (CPM) to take into account local information, using b_t. A final posterior probability distribution $P(\alpha \mid b_t)$ is produced to decide on the action most likely to be happening. The system here described does not consider the temporal extents of the actions and restricts to the current instant t for simplicity. The architecture of the system allows sharing the view independent action models across different views, being able to load them from a common

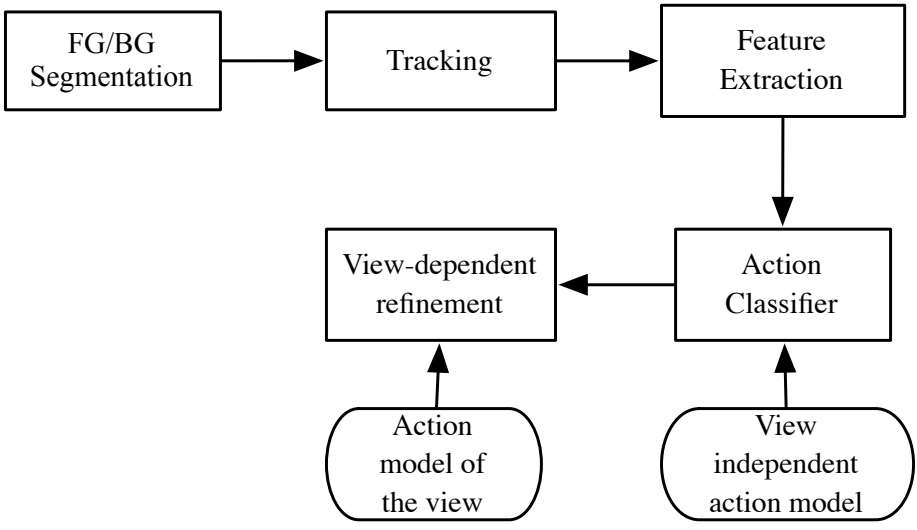


Fig. 1. Overview of the proposed system

library. The system is designed to work with 2D view-independent features, as only the images grabbed from a single camera are used. Also, instead of locating the user in the scene 3D world, it is only located in the 2D view. A similar system could be designed to process different simultaneous views of the scene and compute 3D view-invariant features, such the derived from visual hulls. The location of the user can then be made in the 3D scene world. Note it is not necessary to have multiple views to obtain the 3D location, but we restrict the proposed system to 2D. Including the size of the person bounding box into the context probability distribution might be a naive form of modelling the depth in the scene, as the real size of the person remains constant but not the size of the respective bounding box.

The usage of posterior probability distributions to model the uncertainties makes possible the use of context information. If the view-independent action classifier would produce a crisp output, there would not be any possibility about make an action decision when the output does not agree with the context model. At the same time, a probability distribution can easily relate actions with the places where they are most likely to happen.

3 View-Independent Action Recognition

Producing a view-independent action recognition system is a hard task that is still an open research issue. Some of the efforts already made to accomplish the task were presented on section 1. Before introducing the proposed classifier, it is worth mentioning that it is not a really view-independent classifier. It does

not make any explicit generalization on the viewpoint, just trying to accurately predict the action class of a set of training samples taken from different viewpoints. However, it is enough to show how the performance of the recognition can improve when context knowledge is added.

3.1 Feature Extraction

The combination of optical flow and appearance information has shown to be effective for the recognition of human actions [19]. Optical flow provides information about the current dynamics of the person, while appearance provides information about the current pose. The main problem when computing features for action recognition using the output of an object tracker is the noise, producing random perturbations on the position and size of the bounding box that can generate large variations on feature values.

A good choice in this case can be the usage of histogram feature representations, as they seem robust to perturbations of the bounding box properties. The Histogram of Oriented Optical Flow (HOOF) [2] provides a scale and direction invariant representation of the motion of a target. The Histogram of Oriented Gradients (HOG) [3] is a widely used feature for object detection, providing representation of the object appearance We use the concatenation of both features f_t as the input to the action classifier.

3.2 Action Classification

At was previously mentioned on section 2, the chosen classifier should provide a probabilistic output in order to be able to average the output with the context information. The Relevance Vector Machine (RVM) [18] is a probabilistic model for classification and regression, belonging to the class of sparse kernel methods. RVM automatically selects the basis functions to be used, being very fast to operate because an input pattern need to be only compared with a few samples stored during training. The details on the derivation and implementations of this model are out of the scope of this paper. Readers are encouraged to check [18] for more details. For us, the RVM is a black box that once trained provides an estimation of the probability $p(\alpha_t \mid f_t)$ of an action α_t given the observed feature f_t.

To use the RVM, we need to specify the kernel function to be used to compare two samples x and y. The form of the descriptor introduced in the previous section is an histogram. This motivates us to choose the χ^2 kernel function:

$$K(x,y) = exp\left(-\frac{1}{2}\sum_{b=1}^{B}\frac{(x^b - y^b)^2}{(x^b + y^b)}\right). \quad (1)$$

where x^b and y^b respectively denote the b bin of two histograms x and y to be compared.

4 Probabilistic Context Model

The PCM is defined to model the existing relationship between the actions and the different zones of the view where they are more likely to be observed. A likelihood is associated to each one of the zones of the image to reflect the plausibility of the action happening there using a generative distribution $p(b_t \mid \alpha_t)$. As the value of the function increases, it would be more likely to observe the action α_t at b_t.

There are different choices to model the generative distribution $p(b_t \mid \alpha_t)$. The most simple is to define a gaussian distribution for each action class, but that can be too rigid in the sense that the actions can be observed only on the neighbourhood of a given zone. That is why a gaussian mixture model seems to be a more appropriate choice:

$$p(b_t \mid \alpha_t = a_i) = \sum_{i=1}^{K_c} \pi_i N(\mu_i, \Sigma_i). \tag{2}$$

where π_i corresponds to the weight of the ith component of the mixture, and $N(\mu_i, \Sigma_i)$ denotes the standard gaussian distribution with mean μ_i and covariance matrix Σ_i. The number of K_c mixture components of each class would be empirically determined during training. The parameters of each one of the gaussian mixtures would be estimated from training samples using the standard Expectation-Maximization algorithm [1].

Other possibility is to use a Kernel Density Estimator [1]. This prevents having to choose a priori the number of components in the mixture. Given a set of training samples $X\left(b^1 \ldots b^{N_i}\right)$ of the ith class, the probability density function for the class is defined as:

$$p(b_t \mid \alpha_t = a_i) = \frac{1}{N_i} \sum_{i=1}^{N_i} K\left(b^i - b_t\right). \tag{3}$$

where $K(.)$ can be any kernel function such a gaussian.

The posterior probability distribution on the action provided by the VIC, $p(\alpha_t \mid f_t)$, is combined with the estimation of the probability of each class on the current area $p(\alpha_t \mid b_t)$, to give a final posterior distribution $p(\alpha_t \mid b_t, f_t)$ of the form:

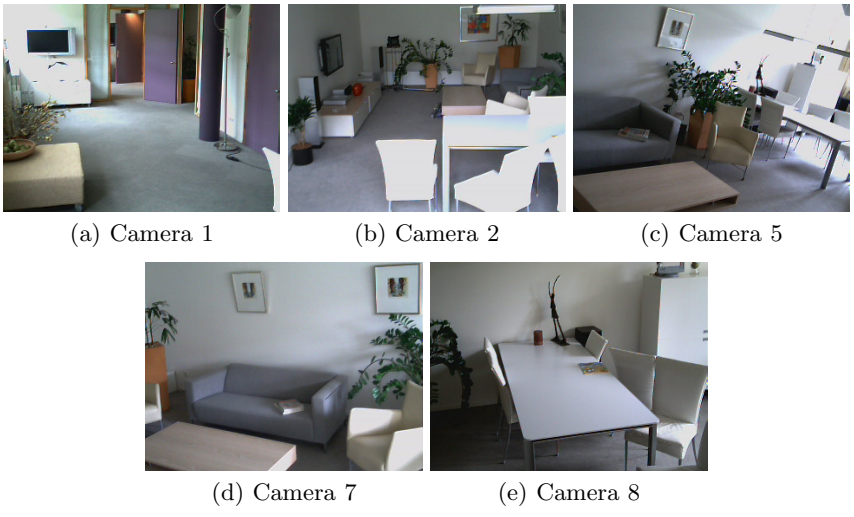
$$p(\alpha_t \mid b_t, f_t) \propto p(\alpha_t \mid b_t)\, p(\alpha_t \mid f_T). \tag{4}$$

It has been assumed that the area b_t and the feature descriptor f_t are independent. $p(\alpha_t \mid b_t)$ is obtained applying the Bayes rule to the generative distributions:

$$p(\alpha_t \mid b_t) = \frac{p(b_t \mid \alpha_t)\, p(\alpha_t)}{p(b_t)}. \tag{5}$$

5 Application Example

Figure 2 some of the different views included in the Philips HomeLab dataset [23]. A living room is shown from different perspectives. The task for this dataset is to recognize when the user is doing different actions, such *reading*, *watching tv*, *walking*, *eating* or *drinking*.



(a) Camera 1 (b) Camera 2 (c) Camera 5

(d) Camera 7 (e) Camera 8

Fig. 2. Different views of the HomeLab dataset

The different actions are observed on most of the views displayed, so the view-invariant classifier can be trained with the action samples grabbed from all the views. Then, for each view, the context model would be learned. This way, for camera 1 it is learned that the action *walk* is mostly performed at the clear space and *sit* close to the *footstool* on the left. For camera 2, that it is very likely to observe the action *walk* on the clear space on the left and *eating*, *drinking* or such related actions close to the table on the right. For camera 5 and camera 7 that close to the sofas is very likely to observe the actions *reading* or *watching tv*. For camera 8 the result would be that *drinking* or *eating* are very likely to be observed in all the image, as the table takes up most of the space. Then, this acquired knowledge would be used to average the probabilities of the action labels most likely to be observed on new action samples.

6 Conclusions and Future Works

This paper has presented and human action recognition system decomposed in two different steps. First, a view-independent classifier, shared between the different views to analyze, provides an initial guess of the action being performed. Then, this estimation is averaged using view-dependent knowledge to make the

final decision on the performed action. This way, the view-independent module can be reused if a new view has to be analyzed, being only necessary to define the view-dependent knowledge.

The system has to be experimentally validated on future works, to quantify how much the use of the context information improves the accuracy of the action classification. We are currently working on obtaining a true view invariant classifier to add to the system. This is done exploiting restrictions derived from simultaneous views of the current action. Other issue to explore is the usage of action models learned from different scenarios, and quantify the plausibility of transferring them to scenarios previously unobserved.

Acknowledgement

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, New York (2006)
2. Chaudhry, R., Ravichandran, A., Hager, G., Vidal, R.: Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 0, 1932–1939 (2009)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2005, vol. 1, pp. 886–893. IEEE, Los Alamitos (2005)
4. Dotu, I., Van Hentenryck, P., Patricio, M., Berlanga, A., García, J., Molina, J.: Real-time tabu search for video tracking association. In: Principles and Practice of Constraint Programming-CP 2009, pp. 21–34 (2009)
5. Gomez-Romero, J., Patricio, M.A., Garcia, J., Molina, J.M.: Context-Based Reasoning Using Ontologies to Adapt Visual Tracking in Surveillance. In: Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009, pp. 226–231. IEEE, Los Alamitos (2009)
6. Gritai, A., Sheikh, Y., Shah, M.: On the use of anthropometry in the invariant analysis of human actions. In: Proceedings of the 17th International Conference on Pattern Recognition ICPR 2004, vol. 2, pp. 923–926 (2004)
7. Lewandowski, M., Makris, D., Nebel, J.C.: View and Style-Independent Action Manifolds for Human Activity Recognition. In: Computer Vision–ECCV, vol. 2010, pp. 547–560 (2010)
8. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and viterbi path searching. In: IEEE Conference on Computer Vision and Pattern Recognition CVPR 2007, pp. 1–8. IEEE, Los Alamitos (2007)
9. Martínez-Contreras, F., Orrite-Uruñuela, C., Herrero-Jaraba, E., Ragheb, H., Velastin, S.A.: Recognizing Human Actions using Silhouette-based HMM. In: IEEE Conference on Advanced Video and Signal-based Surveillance, pp. 43–48 (2009)

10. Parameswaran, V., Chellappa, R.: View invariants for human action recognition. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2003)
11. Peng, B., Qian, G., Rajko, S.: View-Invariant Full-Body Gesture Recognition via Multilinear Analysis of Voxel Data. In: Third ACM/IEEE Conference on Distributed Smart Cameras (September 2009)
12. Poppe, R.: A survey on vision-based human action recognition. Image and Vision Computing 28(6), 976–990 (2010)
13. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. International Journal of Computer Vision 50(2), 203–226 (2002)
14. Richard, S., Kyle, P.: Viewpoint Manifolds for Action Recognition. EURASIP Journal on Image and Video Processing (2009)
15. Robertson, N., Reid, I.: A general method for human activity recognition in video. Computer Vision and Image Understanding 104(2-3), 232–248 (2006)
16. Sheikh, Y., Sheikh, M., Shah, M.: Exploring the space of a human action. In: Tenth IEEE International Conference on Computer Vision ICCV 2005, vol. 1 (2005)
17. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2. IEEE, Los Alamitos (2002)
18. Tipping, M.E.: Sparse Bayesian Learning and the Relevance Vector Machine. Journal of Machine Learning Research 1, 211–244 (2001)
19. Tran, D., Sorokin, A., Forsyth, D.: Human activity recognition with metric learning. In: Proceedings of the 10th European Conference on Computer Vision: Part I, p. 561. Springer, Heidelberg (2008)
20. Turaga, P., Veeraraghavan, A., Chellappa, R.: Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008, pp. 1–8. IEEE, Los Alamitos (2008)
21. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3d exemplars. In: IEEE 11th International Conference on Computer Vision ICCV 2007, pp. 1–7. IEEE, Los Alamitos (2007)
22. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. Computer Vision and Image Understanding 104(2-3), 249–257 (2006)
23. Wu, C., Aghajan, H.: User-centric environment discovery with camera networks in smart homes. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 41(2), 375–383 (2011)

A General Purpose Context Reasoning Environment to Deal with Tracking Problems: An Ontology-Based Prototype

Miguel A. Serrano, Miguel A. Patricio, Jesús García, and José M. Molina

GIAA, Universidad Carlos III de Madrid, Avenida de la Universidad Carlos III, 22, Colmenarejo, Madrid, Spain
{miguel.serrano,miguelangel.patricio,jesus.garcia, josemanuel.molina}@uc3m.es

Abstract. The high complexity of semantics extraction with automatic video analysis has forced the researchers to the generalization of mixed approaches based on perceptual and context data. These mixed approaches do not usually take in account the advantages and benefits of the data fusion discipline. This paper presents a context reasoning environment to deal with general and specific tracking problems. The cornerstone of the environment is a symbolic architecture based on the Joint Directors of Laboratories fusion model. This architecture may build a symbolic data representation from any source, check the data consistency, create new knowledge and refine it through inference obtaining a higher understanding level of the scene and providing feedback to autocorrect the tracking errors. An ontology-based prototype has been developed to carry out experimental tests. The prototype has been proved against tracking analysis occlusion problems.

Keywords: Context Reasoning, Occlusion Problem, Knowledge Approach, Video Tracking.

1 Introduction

The widespread use of video applications in new areas and to new publics has led to an increase amount of raw and unclassified resources. This trend has generated the need to automatically get semantic information by analyzing the resources and cataloging the data to achieve a scene interpretation.

Low informative quality of the data presented by video applications has resulted in an incapability to manage the extracted information on a semantic level. For years researchers have been working in two different ways to save the semantic gap; machine learning methods focused on low level visual descriptors such as texture, shape, etc. and knowledge approaches, which try to extract semantic descriptions using a higher abstraction perspective. None of these approaches can only by itself cover the entire problem.

Increasingly low level and knowledge approaches are being used in a synergistic way taking in account their abstraction level. Machine learning methods are used to imitate the way users assess visual similarity. Knowledge-based approaches are used

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, part II, LNAI 6679, pp. 144–154, 2011.

to discover new hidden domain knowledge through inference operations. However, these mixed approaches are not normally prepared to accept different kinds of measures form different kinds of devices, to fuse all the data generated and to make inference reasoning over these. In addition, normally with these solutions is not possible to deal specific problems at different abstraction levels.

In order to solve these limitations this paper presents an integral environment which combines machine learning methods with a general purpose knowledge approach based on symbolic representation formalism. The goal of this environment is to carry out a comprehensive tracking analysis extracting all the semantics at different abstraction levels from the resources.

The general context reasoning environment is divided in three main modules a tracking system, an annotation system and a knowledge system. The tracking module collects part of the data performing a low level analysis over the raw resources. The annotation module carry out tasks such as annotating and showing the attained and inferred data, controlling the main flow of the execution and acting as interface between the user and the rest of the system. The knowledge module is built under the guidance of an architecture [3] based on a symbolic version of the Joint Directors of Laboratories (JDL) model [4] to deal with tracking problems. The JDL-based architecture is the cornerstone of the overall architecture. Their main capabilities are symbolic representation of the real world with ontologies, information flow refined through deductive and abductive reasoning from low level data to high level domain knowledge, consistency in the data inherent to the chosen representation formalism and feedback information that makes suggestions to improve the behavior of the tracking system.

To test the potential of this environment a prototype has been developed. As a demonstration of the current performance to solve problems at several levels, the system has been tested to the automatic detection and treatment of occlusion situations between scene objects.

The paper is organized as follows. In Section 2 the overall architecture is presented. Section 3 shows the detailed implementation of knowledge model prototype. Section 4 shows an example of how a typical video tracking problem is solved applying the solution presented. Section 5 explains the conclusions obtained and the future work.

2 Overall Architecture

As it is illustrated in Fig. 1, the overall architecture is composed by a tracking module, an annotation module and a knowledge module. A user supervisor is normally necessary to monitor the tracking performance. The basic inputs for this architecture are a data formalism, a variable quantity of predefined data and the video frames. The two first, are inputs of the knowledge module. The data formalism represents the skeleton of concepts, placed according to their abstraction level, where the data is stored. Optionally this formalism could be accompanied with predefined data, such as, assertions of a knowledge base. Normally these data come from previous analysis and context information. The video frames are the tracking module inputs; each frame starts the sequence analysis whose information stream covers all the environment elements.

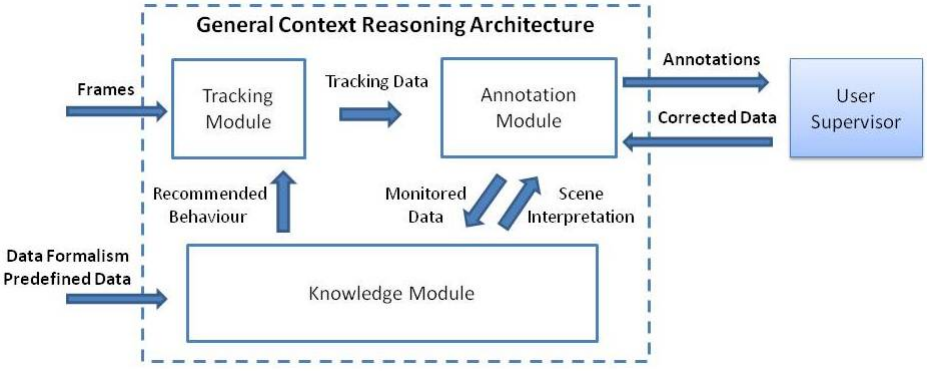


Fig. 1. Overall architecture of the general context reasoning environment

After a frame entry, a real time tracking analysis is performed. When the analysis have finished the tracking module returns information related with tracking entities, such as, position, size, etc. The annotation module supports two inputs the tracking data and the previous frame scene interpretation. The scene interpretations are the semantic conclusions of the knowledge module about the situation of the video scene at different abstraction levels. All these inputs are presented as annotations to the user supervisor. Once the information is checked by the user, the annotation module sends this data monitored to the knowledge module. The knowledge module triggers its reasoning abilities obtaining a new interpretation of the video scene and recommendations of how should the tracking module behave during the analysis.

2.1 Tracking Module

The tracking module architecture, presented in Fig 2, is based on a video chain with different submodules that run in sequence, which correspond to the successive phases of the tracking process.

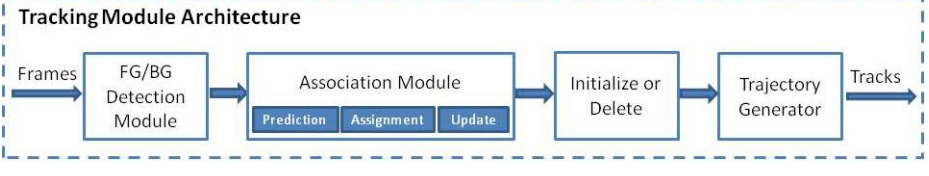


Fig. 2. Tracking Module Architecture

This architecture is composed by four processes: Foreground/Background Detection module, shows when a pixel has moved and group them in blobs. Association module, predicts the blobs positions, assign sets of blobs to tracks and finally update the tracks positions. Initialize or Delete module, create and delete tracks when have not assigned to any blob. Trajectory Generator module, detects anomalous behaviors studying tracks trajectories. The algorithms belonging to each submodule are interchangeable in each run. The input data for the pipeline is a frame and the output data is the track features.

2.2 Annotation Module

The annotation module is based on the ViPER-GT tool [10]. The initial goal of this tool was to provide instruments to easily create and share ground truths in a flexible format. It was designed to allow frame-by-frame markup of video metadata stored in a XML format. The GUI developed is very useful for visualization and could be used to record the requisite information in a single scan of the video content.

The modified version of this tool [7] allows user to supervise data generation through predefined dynamic templates. Predefined templates present the information related with the knowledge module abstraction levels tracking, objects, activities, etc. The annotation tool also offers some possibilities like creation, deletion and update on the semantics of the scene, for example, the user could update in a template the size of a track. All the new data have to be finally sent to the knowledge module due to user updates may modify the interpretation of the global situation. For example, the user could update in a template the size of a context object which modifies the semantic relationships between some other objects.

2.3 Knowledge Module

The knowledge module is an ontology based model compliant with JDL to represent knowledge in cognitive visual data fusion systems [3]. This model is stepped in several levels from low-level track data to high level situations whose structure is determined by a set of ontologies. Each ontology level provides a skeleton that includes general concepts and relations to describe the computer vision entities and their relations.

- TREN ontology (L1). The input data come from the tracking module and represent tracked entities information (color, position, velocity, etc.) and frames.
- SCOB ontology (L1-L2). Represents real-world entities, properties and relations of the scene: moving and static objects, topological relations, etc.
- ACTV ontology (L2). Models the behavior descriptions: grouping, approaching, picking an object, etc.
- IMPC ontology (L3). Define the association of a cost value to activity descriptions.

The ontology model has been designed to promote extensibility and modularity. This general structure might be refined to apply this model in a specific domain. It is interesting to note that the amount of data in lower level ontologies is larger than higher levels. Concepts that belong to less abstract knowledge form the basis of the high abstract knowledge.

These ontologies can contain both perceptual and context data. Perceptual data is automatically extracted by the tracking algorithm. Context data is external knowledge used to complete the comprehension of the scene. Context data includes information about scene environment, the parameters of the recording, information previously computed and user-requested information. For example, an static object and its features size, position, topological state, etc. are regarded as context data. The

producing sources of context are the a priori knowledge and the users. A priori knowledge is generally common sense data introduced before the initialization of the system. This information is usually closely linked to the specific purpose of the application. Users must also provide the required amount of the context data interacting with the annotation module. This required data should be introduced during the execution when a bad comprehension of the scene is detected in the behavior of the tracking module.

New knowledge assertions in each ontology from the proper and the previous level are also possible thanks to the abductive reasoning. Beyond the standard reasoning based on the subsumption ontology mechanism, this model can perform rule based inferences using a description logic inference engine. Deductive and abductive rules are the two main types of reasoning procedures. Deductive rules maintain the consistency in the ontology. The consequent include logical conditions and not include in the consequent any individual not mentioned in the antecedent. Abductive rules are used to achieve the scene interpretation and creation of feedback for the tracking module. These rules allow the inclusion of additional individuals in the consequent, which are created as new instances of the ontology. Abductive reasoning use additional mechanism like data retrieval queries, previously stored queries or queries depending on arithmetical calculus.

3 Ontology Based Prototype: Knowledge System

The ontology based prototype implements the overall architecture seen in section 2, however, in terms of research quality the key piece of this architecture is the knowledge module. This module must carry out the fundamental tasks such as asserting symbolic context data, creating a symbolic representation of these data, checking the data consistency, making deductive and abductive inference and providing feedback at different levels.

As is illustrated in Fig. 3, the developing of this module is almost entirely based on the RACER reasoner [9]. RACER has been selected because of the necessity of the system to develop different kinds of reasoning throw rules capable to infer new knowledge. In addition, RACER is one of the first reasoners to treat the spatial knowledge with an implementation of the Region Connection Calculus (RCC) theory.

The knowledge module is represented by the computer vision symbolic representation (CVSR). The actual CVST implementation includes the three first lowest levels of the architecture, the tracking entities level (TREN), the scene object level (SCOB) and the activities level (ACTV). These levels are defined from a set of terminological box (TBox) that contains the hierarchical description of the concepts defining the classes and relations of the ontology of each level. CVSR assertional boxes (ABoxes) indicate the assertions about the individuals in the domain. The consistency verify whether all concepts in the TBox admit at least one individual in the corresponding ABox. These individual assertions come from predefined context knowledge given by the user or previous executions. Currently the system loads all

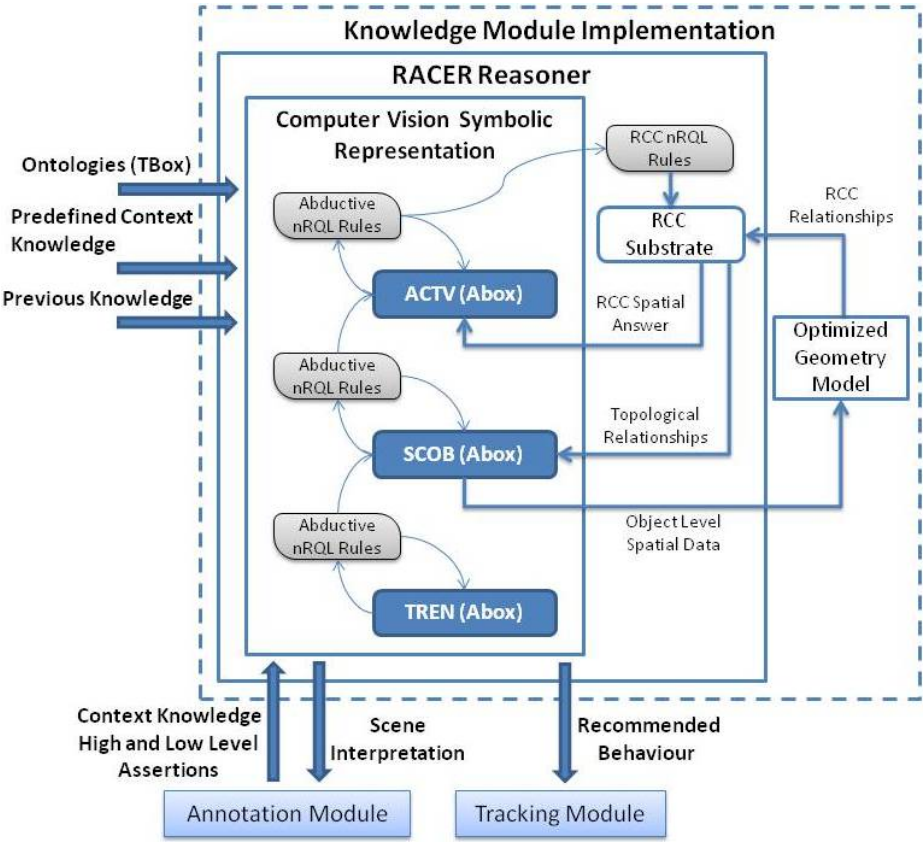


Fig. 3. The knowledge module implementation of the ontology based prototype

these data, prior to the start of the execution, from a file previously saved with information of an ABox.

During the execution, more context knowledge could be asserted by the user through the annotation system interface or by abductive new Racer Query Language (nRQL) rules. nRQL can be used to query RACER ABoxes, TBoxes and substrates and RDF and OWL documents. Abductive rules make flow new and more elaborated data at the same level or from low to upper abstraction levels. In the example below it can be seen a rule that generate new upper knowledge transforming a track in a dynamic scene object of the person class. This rule makes use of some reasoning facilities previously explained. Rule antecedent contains perceptual data such as track identifiers, previously stored queries like current-snapshot function, the a priori context knowledge showed in the width and height track properties condition, etc. Rule consequent generates a new individual labeled as person, declares a specific identifier and creates a new role which associates the individual to a track entity.

```
(firerule
  (and
    (?track (equal #!tren:id trackID)
    (?track ?tsnapshot ?dimension current-snapshot)
    (?tsnapshot #!tren:unknown_frame #!tren:isValidInEnd)
    (?dimension (>= #!tren:width 9))
    (?dimension (>= #!tren:height 9)))
  (
    (instance (new-ind |ind|) #!scob:Person)
    (instance (individual |ind|)(equal #!scob:id trackID))
    (related (individual |ind|)?track #!scob:hasAssociatedTrack)))
```

Abductive reasoning with spatial objects is very expensive in terms of development of rules and time computing and grows with the increasing number of entities and the complexity of the scene. An ontology-centric architecture based on an optimized geometry model [8] has been designed to solve the topological relationships between spatial objects automatically. The capability for automatic assertion is given by an object model based on geometries. The object model seeks to prioritize the optimization using a dynamic data structure of spatial data. The main goal is the automatic storage of the spatial relationships on a RCC without a noticeable loss of efficiency.

RCC is a theory used for qualitative spatial representation and reasoning. This theory provides a formalism which allows inferring implicit knowledge from explicit knowledge and a set of qualitative spatial relationships to describe the relative location of spatial objects to one another. The RCC theory is implemented by RACER as a substrate. In general substrate representation layers are used to associate a RACER ABox with an additional representation layer. The RCC substrate is a special kind of data substrate used to represent domain objects that also have spatial characteristics. This substrate also offers querying facilities, nRQL support both spatial and combined spatial and non-spatial queries. A significant amount of knowledge from the scene object and activities level is deducted from rules that manage the spatial object relationships discovered by the optimized geometries model.

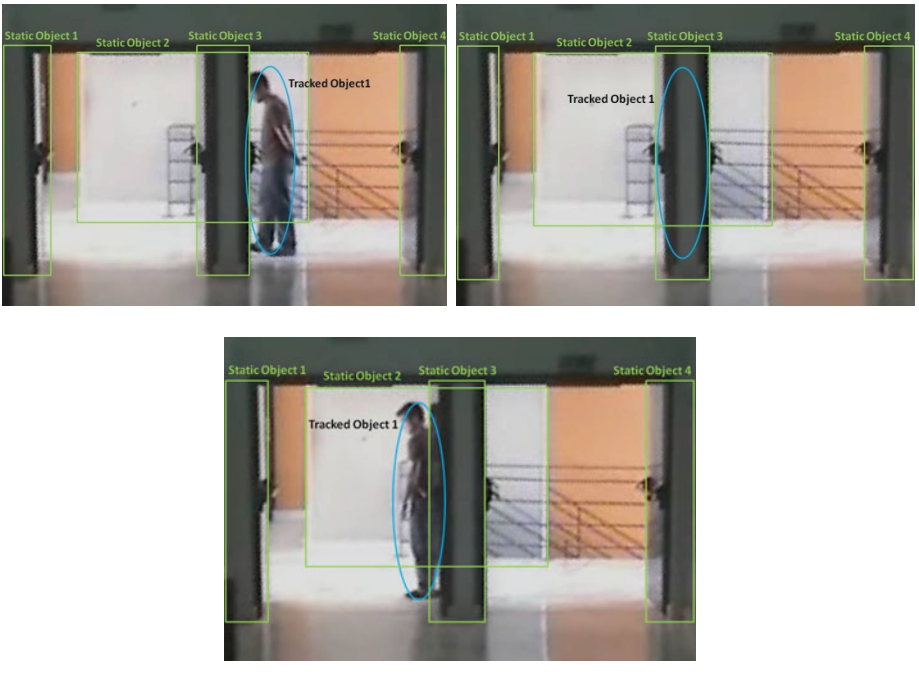
The scene interpretation output is presented to the user at the predefined dynamic templates of the annotation system. This allows the user to edit the erroneous semantic conclusions of the knowledge module at different abstraction levels to improve further recommendations. The tracking system recommendations have not been fully adapted to the prototype. The architecture presented in [1] and [2] uses a specific ontology called RECO which abstractly represents actions to improve the accuracy of the tracking system, and consequently, the quality of the measures of the fusion system. The current development directly modifies the features of the tracks at the lowest level to change the tracking behavior.

4 Case Study: Occlusion Problem

In this case study the main aim is to recognize and treat occlusion situations. The system has been tested against videos which show a scenario with several columns.

These columns have been defined by the user supervisor as context information and more specifically as a column concept which may be also considered as a static objects because one is a less general than the other.

Table 1. Illustrated sequence of how the overall system treats the occlusion. 1A shows the beginning of an occlusion. 1B shows the track completly occluded by the column. 1C image shows the occlusion ending.



The information given from the user to the system was the position, the size and the definition of the columns. The tracking system automatically detects the track position at every moment. This track is asserted in the TREN ontology with its features. The knowledge module detect that the track is a person and creates an individual in SCOB to represent the track at the object level from the rule view in section 3. Thus the system can make inference on the same level with columns and tracks.

An occlusion is detected when a tracked object in an overlap relation with a static object, in this case a column, is deleted by the tracking system. In 1A the prototype does not confuse as an occlusion the situation between the “static object 2” and the “tracked object 1” because the tracked object is not deleted by the tracking system. To recognize and assert occlusion situations is necessary the use of inference rules with tracking entities and scene objects in the antecedent and activities in the consequent.

```
(firerule
  (and
    (?track #!tren:Track)
    (?object #!scob:StaticObject)
    (?track (equal #!tren:id trackID))
    (?track ?person #!scob:isAssociatedToObject)
    (?*object ?*track :po))
  (
    (instance (new-ind |ind|) #!actv:Occlussion)
    (related (individual |ind|) ?object #!actv:hasOccluder)
    (related (individual |ind|) ?person #!actv:hasOccluded))
```

Table 2. Knowledge management sequence corresponding to the the image sequence in table 1

	Image 1	
Previous Knowledge	TREN	**Instances:** Track_1 [position and size]
	SCOB	**Instances**: Static_object1, Static_object2, Static_object3, Static_object4 [positions and sizes]
Rule fired	Track_1 matches the person rule, new person instance is created.	
Rule Asserted Knowledge	TREN	**Relationships:** Track_1 isAssociatedTo Object PersonObj1
	SCOB	**Instances:** PersonObj_1 **Relationships:** PersonObj_1 hasAssociatedTrack Track_1
Tracking Event	Track_1 deleted.	
Knowledge Event	Track_1 and PersonObj_1 declared as inactive in knowledge base. Overlapping rule fired.	
Image	**Image 2**	
Previous Knowledge	SCOB	**Instances:** Static_object1, Static_object2, Static_object3, Static_object4 [positions and sizes]
Rule fired	PersonObj_1 matches the overlapping rule.	
Recommended Asserted Knowledge	TREN	**Instances:** Track_1 [new position, old size] **Relationships:** Track_1 isAssociatedTo Object PersonObj1
	SCOB	**Instances:** PersonObj_1 **Relationships:** PersonObj_1 hasAssociatedTrack Track_1
Rule Asserted Knowledge	ACTV	**Instances:** Occlusion_1_3 **Relationships:** Occlusion_1_3 hasOccluder Static_object2 Occlusion_1_3 hasOccluded PersonObj1
Image	Image 3	
Previous knowledge	TREN	**Instances:** Track_1 [position, size] **Relationships:** Track_1 isAssociatedTo Object PersonObj1
	SCOB	**Instances:** PersonObj_1, Static_object1, Static_object2, Static_object3, Static_object4 [positions and sizes] **Relationships:** PersonObj_1 hasAssociatedTrack Track_1
	ACTV	**Instances:** Occlusion_1_3 **Relationships:** Occlusion_1_3 hasOccluder Static_object2 Occlusion_1_3 hasOccluded PersonObj1

When the occlusion is fully detected some recommended behavior are sent directly to the tracking system prioritizing the knowledge system recommendations before the tracking algorithm detections. For the situation seen in 1B image the recommendations are re-create the track, maintain the last registered size before the deletion and situate the object centered in the object that occludes. The tracked object is automatically re-activated through the rule seen in Section 3. Table 2 shows a simplified view of how the specific data of the occlusion situation flow from low level to high level comprehension and scene interpretation.

5 Conclusion and Future Work

We have presented a general purpose approach based context knowledge and inference reasoning to deal with general and specific tracking problems at different fusion levels. An ontology based prototype has been developed to demonstrate the performance of this knowledge module. The system has been tested against the occlusion problem.

Future works will be addressed to expand the number of JDL model levels implemented. This extension will include ontologies to manage the impact of the recognized activities and the recommendations from the knowledge module to the tracking module. Therefore new rules, for abductive and deductive reasoning for these levels will have to be developed.

Acknowledgement

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/ TIC-1485) and DPS2008-07029-C02-02.

References

1. Gómez-Romero, J., Patricio, M.A., García, J., Molina, J.M.: Towards the implementation of an ontology-based reasoning system for visual information fusion. Applied Artificial Intelligence Group, Departament of Computer Science, Universidad Carlos III de Madrid (2009)
2. Gómez-Romero, J., Patricio, M.A., García, J., Molina, J.M.: Context-based reasoning using ontologies to adapt visual tracking in surveillance. In: Advanced Video and Signal Based Surveillance, pp. 226–231 (2009)
3. Gómez-Romero, J., Patricio, M.A., García, J., Molina, J.M.: Ontological representation of context knowledge for visual data fusion. In: 12th International Conference on Information Fusion, Ontological representation of context knowledge for visual data fusion, pp. 2136–2143 (2009)
4. Llinas, J., Bowman, C., Rogova, G., Steingber, A., Waltz, E., White, F.: Revisiting the JDL data fusion model II. In: 7th International Conference on Information Fusion, Stockholm, Sweden, pp. 1218–1230 (2004)

5. Randell, D.A., Cui, Z., Cohn, A.G.: A Spatial Logic based on Regions and Connection. In: Proceedings 3rd International Conference on Knowledge Representation and Reasoning (1992)
6. Sánchez, A.M., Patricio, M.A., García, J., Molina, J.M.: A Context Model and Reasoning System to Improve Object Tracking in Complex Scenarios. In: Expert Systems with Applications, vol. 36, pp. 10995–11005 (2009)
7. Serrano, M.A., Gracía, J., Patricio, M.A., Molina, J.M.: Interactive video annotation tool. In: Carvalho, A.P.L.F.D., Rodríguez-González, S., De Paz Santana, J.F., Rodríguez, J.M.C. (eds.) Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing, vol. 79, pp. 325–332. Springer, Heidelberg (2010)
8. Serrano, M.A., Patricio, M.A., García, J., Molina, J.M.: Dynamic RCC Automatically Updated for Topological Reasoning. In: Distributed Computing and Artificial Intelligence (2011) (accepted)
9. RACER. Renamed ABox and concept expression reasoner, `http://www.racer-systems.com/`
10. The video performance evaluation resource ground truth authoring tool (ViPER-GT), `http://viper-toolkit.sourceforge.net/products/gt/`

Accuracy Updated Ensemble for Data Streams with Concept Drift

Dariusz Brzeziński and Jerzy Stefanowski

Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 2, 60–965 Poznań, Poland
{dariusz.brzezinski,jerzy.stefanowski}@cs.put.poznan.pl

Abstract. In this paper we study the problem of constructing accurate block-based ensemble classifiers from time evolving data streams. AWE is the best-known representative of these ensembles. We propose a new algorithm called Accuracy Updated Ensemble (AUE), which extends AWE by using online component classifiers and updating them according to the current distribution. Additional modifications of weighting functions solve problems with undesired classifier excluding seen in AWE. Experiments with several evolving data sets show that, while still requiring constant processing time and memory, AUE is more accurate than AWE.

1 Introduction

Ensembles have attracted many researchers as an approach for improving predictive accuracy. However, most of the research is devoted to static environments where the classification task is fixed and complete data is available for learning classifiers. On the other hand, a new type of problems is becoming more visible, one in which learning algorithms work in *dynamic environments* with data continuously generated in the form of a stream [1]. Processing *data streams* implies new requirements concerning limited amount of memory, small processing time, and one scan of incoming data. Moreover, the data distributions and definitions of target classes change over time. These changes are categorized into *sudden* or *gradual concept drift* depending on appearance of novel classes in a stream and the rate of changing definitions of classes [2]. Concept drifts directly influence algorithm classification abilities as classifiers generated prior to change have been trained on a different class distribution. As the reason of these changes is hidden and not known a priori, the task of learning classifiers becomes very difficult.

A classifier (individual or ensemble), if intended for such *non-stationary environments*, has to adapt to concept drifts. Several adaptation methods have been proposed including mainly: sliding window approaches, new online algorithms, special detection techniques, and adaptive ensembles. In the area of *adaptive ensembles*, component classifiers are generated from sequential blocks of training examples. When a new block arrives, classifiers are evaluated and later updated, removed, or modified according to the result of the evaluation. *Accuracy Weighted Ensemble* (AWE) is the most popular method in this area [3]. However,

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 155–163, 2011.

defining an appropriate size of the data block can be problematic. Moreover, too many component classifiers can be excluded from the ensemble when they are not accurate enough. We have also noticed in preliminary experiments that AWE is not as accurate as other online classifiers [4].

Therefore, we decided to propose a new algorithm, called *Accuracy Updated Ensemble* (AUE), which would improve over AWE on classification accuracy, while still keeping good computational efficiency. The other aim of this paper is to evaluate the proposed algorithm on several changing data sets and compare it with AWE and other related ensembles available in the MOA framework [5].

2 Related Work

Similarly to most researchers we consider completely supervised learning, i.e. class labels of incoming examples in the stream are available and can be used for evaluating and updating a classifier. Below, we briefly discuss methods most related to our proposal and those used in experiments. For more comprehensive reviews, in particular concerning online incremental ensembles consult [1,2]. Unlike ensembles that can be modified after reading single examples, we discuss streams divided into *blocks of examples* (non-overlapping and of the same size). In an adaptive ensemble, each component is learned using typical "batch" mode on the most recent block (also called a *chunk*) of data. Following discussion from [1], in time-changing streams, data is generated from a mixed distribution, which can be seen as a weighted combination of distributions characterizing the target concepts. This justifies multiple classifiers where each component classifier receives a *weight* reflecting its performance on the most recent block of data.

The SEA algorithm [6] was one of the first algorithms following this idea of building separate classifiers from sequential blocks. Component classification scores are evaluated on the newest block and the weakest classifier in the fixed size ensemble can be replaced by a newly trained one.

Another way of restructuring an ensemble was proposed by Wang et al. [3]. In their *Accuracy Weighted Ensemble* (AWE), the authors propose to train a new classifier on each incoming data block and use that block to evaluate all the existing classifiers in the ensemble. They formally proved that if component classifiers are weighted by their expected accuracy on the test data, the ensemble improves classification accuracy over a single classifier. To weight the members of an ensemble we need to know the actual function being learned, which is unavailable. That is why Wang et al. proposed to derive weights by estimating the error rate on the most recent data block x_i, as shown in Equations 1-2:

$$MSE_i = \frac{1}{|x_i|} \sum_{(x,c) \in x_i} (1 - f_c^i(x))^2, \quad MSE_r = \sum_c p(c)(1 - p(c))^2, \tag{1}$$

$$w_i = MSE_r - MSE_i, \tag{2}$$

where $f_c^j(x)$ denotes the probability given by classifier C_i that x is assigned to class c. The value of MSE_r is the mean square error of a randomly predicting

classifier and it is used as a threshold to zero the weights of classifiers, which are not accurate enough. For the first k data chunks AWE takes a set of all available classifiers, but when processing further chunks it selects only the k best components to form an ensemble. For a large data stream it is impossible to remember all the components and the number of stored components should be limited. The predictions of components are aggregated by a weighted voting rule. Experiments comparing AWE with different single classifiers, such as C4.5, RIPPER, and Naive Bayes showed that AWE improved both computational efficiency in learning and classification accuracy.

The performance of AWE and other block-built ensembles largely depends on the size of the data blocks. Bigger blocks can lead to more accurate classifiers, but can contain more than one concept drift. On the other hand, smaller blocks are better at separating changes, but usually lead to poorer classifiers.

In our approach we learn classifiers with *Very Fast Decision Trees* (also known as *Hoeffding Trees*). They were introduced in [7] to efficiently learn from massive data in an incremental way without the need for storing consecutive examples. The tree is learned by recursively replacing leaves with decision nodes. Each leaf stores sufficient statistics about attribute values, which are needed by an evaluation function that judges the merit of split-tests based on attribute values. The key idea is to show that a relatively small sample can be enough to choose the optimal split-test, based on collected statistics. The main innovation is the use of the *Hoeffding bound* to guarantee that the selected split is really the best one. The original VFDT algorithm was proposed for static data streams. However different adaptations to handle concept drift were later proposed; see their review in [1].

VFDT are also used in an ensemble called *Option Trees*. This generalization of a single tree includes *option nodes* where instead of selecting only the best split-test attribute, all promising attributes are kept. Later, for each of those attributes a decision subtree is constructed. Thus, making a final decision with an option tree involves weighted combining of the predictions of all applicable subtrees. In our experiments we use Kirkby's [8] proposal of Hoeffding Option Trees (HOT) for streaming data (see section 10.4 in [1] for details).

3 Accuracy Updated Ensemble

Following the critical discussion of AWE in Section 1, we propose a new adaptive ensemble called *Accuracy Updated Ensemble* (AUE). The proposed algorithm is inspired by AWE and its weighting mechanism, but improves its flaws. AUE not only selects classifiers, but also *updates* them according to the current distribution. Let us remind that processing of data blocks allowed AWE to learn component classifiers by "traditional batch" algorithms (not special online ones) and later only adjust component weights according to the current distribution. However, this leads to problems with tuning the block size (in [3] it was estimated in many trails). In AUE we turn to online learning of component classifiers. This allows to update base classifiers rather than only adjust their weights. If no change

occurs between a series of blocks, the component classifier will improve just as if it was built on a bigger block (more appropriate for periods of stability). As a result, we can reduce the size of the block without the risk of creating less accurate components. Furthermore, we preserve the basics of AWE's weighting mechanism and depreciate classifiers if sudden drift occurs. The combination of classifier selection and updating should make AUE better than AWE in times of stability or gradual drift, while being at least as accurate for sudden drift.

Another drawback of AWE is its weighting function. Because the algorithm is designed to perform well on cost-sensitive data [3], the MSE_r threshold in Equation 2 cuts-off "risky" classifiers. In rapidly changing environments with sudden concept drifts (as the Electricity data set) this threshold can "mute" all ensemble members causing no class to be predicted. To avoid this, in AUE we propose a simpler weighting function:

$$w_i = \frac{1}{(MSE_i + \epsilon)} \tag{3}$$

MSE_i is calculated just like in Equation 1 and ϵ is a very small constant value, which allows weight calculation in rare situations when $MSE_i = 0$.

We want to update component classifiers according to the current distribution, while still keeping their diversity. To achieve this, we update only selected classifiers. First of all, we consider only current ensemble members - the k top weighted classifiers. Then we use MSE_r as a threshold for allowing online updating of only "accurate enough" classifiers (line 12 of AUE psuedo-code). Therefore, inaccurate classifiers can enter the ensemble, but will not be updated. The full pseudo-code of the Accuracy Updated Ensemble is listed in Algorithm 1.

Algorithm 1. Accuracy Updated Ensemble

Input: $\mathcal{S}$: data stream of examples
k: number of ensemble members
Output: $\mathcal{E}$: ensemble of k online classifiers with updated weights

```
1: C ← ∅;   //C: set of stored classifiers
2: for all data chunks x_i ∈ S do
3:    train classifier C' on x_i;
4:    compute error MSE of C' via cross validation on x_i;
5:    derive weight w' for C' using (3);
6:    for all classifiers C_i ∈ C do
7:       apply C_i on x_i to derive MSE_i;
8:       compute weight w_i based on (3);
9:    E ← k of the top weighted classifiers in C ∪ {C'};
10:   C ← C ∪ {C'};
11:   for all classifiers C_e ∈ E do
12:      if w_e > 1/MSE_r and C_e ≠ C' then update classifier C_e with x_i;
```

To sum up, AUE differs from AWE in the definition of the weight function, the use of online base classifiers, and updating components with incoming examples.

Ensemble members are weighted, can be removed, and are not always updated, unlike in online bagging [1]. Compared to VFDT based ensembles, ASHT and HOT, we do not limit base classifier size, do not use any windows, and update members only if they are accurate enough according to the current distribution.

4 Experimental Evaluation

The aim of the experiments was to compare the newly proposed AUE with AWE and two other stream classifiers: the Hoeffding Option Tree (HOT) and a single Hoeffding Tree with a static window (HT+Win). We chose AWE as it is the classifier we tried to improve, HOT as a different ensemble that uses Hoeffding Trees, and HT+Win as a reference point to using a single classifier. In all the compared algorithms we construct component classifiers with Hoeffding Trees and compare basic characteristics on popular synthetic and real life data sets.

All of the tested algorithms were implemented in Java as part of the MOA framework [5]. We implemented the AWE and AUE algorithms, and a data chunk evaluation procedure, while all the other algorithms were already a part of MOA. The experiments were done on a machine equipped with an Intel Pentium Core 2 Duo P9300 @ 2.26 GHz processor and 3.00 GB of RAM. To make the comparison more meaningful, we set the same parameter values for all the algorithms. For ensemble methods we set the number of component classifiers to 15: AWE and AUE have 15 trees, HOT has 15 options. The size of data block, as suggested in [3], is equal $d_r = 500$ and $d_a = 1000$, for real and artificial data sets, respectively. We also set the static window size to $15 \times blockSize$ to make the number of examples seen by the windowed classifier similar to that seen by AWE and AUE. The parameters of the Hoeffding Tree used with a static window are the same as those of the option tree, and the base classifiers (also Hoeffding Trees) of the ensembles.

According to the main characteristics of data streams [5,4], we evaluate performance of algorithms with respect to time efficiency, memory usage, and accuracy. Classification accuracy was calculated using the *data block evaluation method*, which works similarly to *test-then-train* paradigm. This method reads incoming examples without processing them, until they form a data block of size d. Each new data block is first used to test the existing classifier, then it updates the classifier. More details concerning this method can be found in [4].

4.1 Data Sets

Following literature on adaptive ensembles we selected three real and four synthetic data sets with concept drift, all of which are publicly available.

Real Data Sets. We selected: Electricity market data (Elec), Ozone level detection (Ozone), and Donation data (Don). Electricity [9] is a data set that consists of energy prices from the electricity market in the Australian state of

New South Wales. Ozone problem concerns local ozone peak prediction, that is based on eight hours measurement [10]. The true model behind the data evolves gradually over time. Another difficulty in mining this data set, is that many of the 72 features collected for each instance are irrelevant. Finally, Donation [11] is a data set used for the 2nd KDDCup. Donation is the largest real data set and contains examples of sudden drift.

Synthetic Data Sets. For testing algorithms on larger data sets we used four popular generators: LED (Led), Waveform (Wave) [1], Hyperplane (Hyp) [3], and SEA (Sea) [6]. LED consists of a stream of 24 binary attributes, 17 of which are irrelevant, that define the digit displayed on a seven-segment LED display. We generated 1,000,000 examples with sudden and gradual concept drift. Waveform consists of a stream with three decision classes where the instances are described by 40 attributes. With Waveform we generated 1,000,000 instances with gradual concept drift. Hyperplane is used to generate a stream of 5,000,000 examples representing gradual concept drift by rotating the decision boundary for each concept. Finally, we use SEA to generate a data stream of 20,000,000 instances with 10% of noise and sudden concept drift.

4.2 Results

Time Analysis. Tables 1 and 2 present average train and test times for each data set. Training time for HOT is usually the highest, in particular for larger artificial data. Of course the single tree (HT+Win) learns much faster than ensembles. We also analyzed graphical plots of time with respect to the number of processed examples[1], which showed that testing times remain close to constant throughout the whole processing of the data stream. This observation is true for all tested data sets. On the other hand, training time is not constant for all algorithms - HOT shows clear linear growth of training time when no sudden drift occurs. Hoeffding trees become more complex as they see more examples. In periods of stability HOT will successively grow bigger trees, thus consuming more memory. The windowed tree, AWE, and AUE are built from a limited number of data blocks, which naturally restricts memory.

Table 1. Average test and train times in ms for data blocks in real data sets

	Elec		Ozone		Don	
	Train	Test	Train	Test	Train	Test
HOT	101.11	6.64	62.40	1.00	2168.54	17.19
AWE	56.91	13.29	179.40	27.30	3290.93	1074.98
AUE	75.11	15.31	241.80	54.60	3292.64	1086.82
HT+Win	**23.98**	**0.29**	**46.80**	**0.00**	**81.10**	**1.76**

[1] Due to space limitations, in this paper we mostly present tabular summaries. For a complete set of plots see [4].

Table 2. Average test and train times in ms for data blocks in artificial data sets

	Led		Wave		Hyp		Sea	
	Train	Test	Train	Test	Train	Test	Train	Test
HOT	563.68	**9.93**	2573.86	51.27	5170.84	14.18	4876.41	6.44
AWE	751.40	181.85	558.21	159.19	41.13	4.96	29.44	5.61
AUE	803.93	185.39	636.59	162.06	240.22	49.39	90.05	18.84
HT+Win	**54.42**	10.81	**36.42**	**6.96**	**17.61**	**3.40**	**6.32**	**1.31**

Memory Usage. According to Table 3, HT+Win used the least memory on most data sets. Additionally, the analysis of all memory plots showed that memory requirements are similar to training time requirements. HOT needs much more memory for larger data sets than HT+Win, AWE, and AUE, which processed the data streams using constant memory. An example of linear growth of HOT's memory requirements is shown in Figure 1.

Table 3. Average size of the classifier for all data sets, measured in MB

	Elec	Ozone	Don	Led	Wave	Hyp	Sea
HOT	0.41	**0.08**	13.97	2.76	12.27	18.49	24.45
AWE	0.23	0.28	5.52	**0.58**	**0.33**	**0.17**	0.18
AUE	0.36	0.36	5.64	0.88	0.92	0.86	0.46
HT+Win	**0.22**	0.30	**1.15**	0.73	0.42	**0.17**	**0.07**

Classification Accuracy. Table 4 presents average classification accuracies obtained by the tested algorithms on all the data sets.

Table 4. Average accuracy for all data sets in percent

	Elec	Ozone	Don	Led	Wave	Hyp	Sea
HOT	74.37	**91.60**	**94.35**	70.68	**82.57**	**85.07**	**89.81**
AWE	71.22	67.59	**94.35**	71.16	79.63	70.38	78.52
AUE	**74.92**	76.56	**94.35**	**71.41**	82.26	84.72	88.61
HT+Win	42.43	**91.60**	**94.35**	60.22	75.46	79.08	87.64

One can notice that none of the tested classifiers is best for all the data sets. For larger data sets, with little and mostly gradual concept drift, HOT gives the best results with AUE being close second. For smaller data sets it is hard to choose a winner. However, in case of sudden changes (Elec, Led) AUE seems to be the most accurate. It is also important to notice that AUE is more accurate than AWE on all data sets (except for Don, where they are equal). The difference in accuracy is especially visible for data sets with gradual drift (Hyp, Wave), where AWE falls far below AUE.

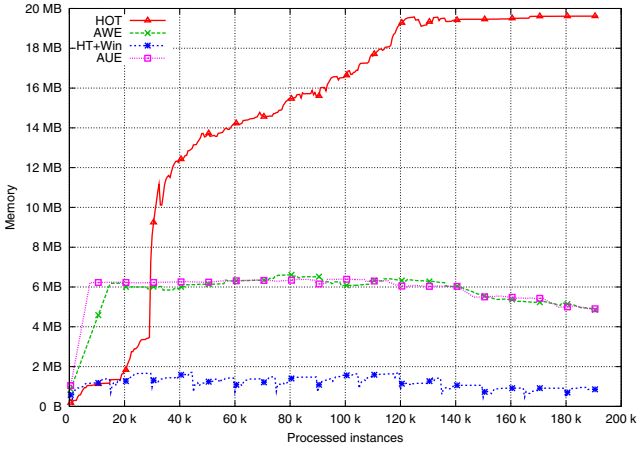


Fig. 1. Growth of HOT's memory requirements (Donation data set)

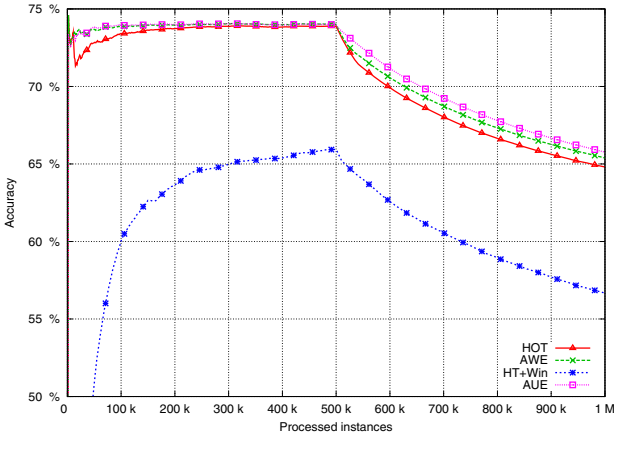


Fig. 2. Accuracy on the Waveform data set

The most interesting concept drift was introduced in the generation of the LED data set. We joined two gradually evolving LED data sets with a sudden change. After half million examples we replaced one data source with another. The algorithms' reactions to this type of change are presented in Figure 2. The plot shows that for this complex concept drift all algorithms have problems with adjusting to change. AUE seems to cope best with this situation. HOT, which performed well before the drift, falls down even below the level of AWE. In periods of stability, HOT grows accurate but complex structures, which are later difficult to rebuild. AWE and AUE are modular, allow quick substitution of components, and therefore quickly react to sudden drifts.

5 Conclusions and Future Work

In this paper we introduced a new stream classifier called Accuracy Updated Ensemble, inspired by an earlier proposed algorithm called Accuracy Weighted Ensemble. AUE was more accurate than AWE on all data sets (except Donation where it was equal) whilst still requiring constant processing time and memory.

Considering the updating technique of AUE we can suspect that in periods of longer distribution stability, when no concept drift occurs, the component classifiers can be trained on more examples and should become more accurate. However, updating many components with similar examples may reduce their diversity. Therefore, in on-going research we study this problem and possible modifications of AUE to ensure additional diversity of ensemble components.

References

1. Gama, J.: Knowledge Discovery from Data Streams. CRC Press, Boca Raton (2010)
2. Kuncheva, L.I.: Classifier ensembles for changing environments. In: Roli, F., Kittler, J., Windeatt, T. (eds.) MCS 2004. LNCS, vol. 3077, pp. 1–15. Springer, Heidelberg (2004)
3. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Getoor, L., Senator, T.E., Domingos, P., Faloutsos, C. (eds.) Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24-27, pp. 226–235. ACM, New York (2003)
4. Brzezinski, D.: Mining data streams with concept drift. Master's thesis, Poznan University of Technology, Poznan, Poland (2010), http://www.cs.put.poznan.pl/dbrzezinski/publications/ConceptDrift.pdf
5. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: Massive online analysis. Journal of Machine Learning Research 11, 1601–1604 (2010)
6. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 26-29, pp. 377–382. ACM, New York (2001)
7. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, August 20-23, pp. 71–80. ACM, New York (2000)
8. Kirkby, R.: Improving Hoeffding Trees. PhD thesis, Department of Computer Science, University of Waikato (2007)
9. Harries, M.: Splice-2 comparative evaluation: Electricity pricing. Technical report, The University of South Wales (1999)
10. Zhang, K., Fan, W., Yuan, X., Davidson, I., Li, X.: Forecasting skewed biased stochastic ozone days: Analyses and solutions. In: ICDM, pp. 753–764. IEEE Computer Society, Los Alamitos (2006)
11. Fan, W.: Systematic data selection to mine concept-drifting data streams. In: Kim, W., Kohavi, R., Gehrke, J., DuMouchel, W. (eds.) Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, pp. 128–137. ACM, New York (2004)

Classifier Ensembles for Virtual Concept Drift – The DEnBoost Algorithm

Kamil Bartocha and Igor T. Podolak

Institute of Computer Science, Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6, Kraków, Poland
kamil.bartocha@uj.edu.pl, podolak@uj.edu.pl

Abstract. Virtual concept drift is a phenomenon frequently arising in applications of machine learning theory. Most commonly, a discard-retrain strategy is the only option for dealing with newly generated data coming from previously unknown areas of the input space. This paper proposes a method of constructing classifier ensembles based on a measure of observations' density and homogeneity of their corresponding labels. The strategy allows to incrementally add new data points into the model without the necessity of a full retraining procedure.

1 Introduction and Problem Statement

In many applications of the machine learning theory, the underlying concept is assumed not to change with time. A model is trained using a sample of data and is expected to perform well in the future. If, in the course of time, new observations arrive from unseen areas of the input space, the model may quickly become obsolete. In recent years, an ensemble based approach to dealing with virtual concept drift has become popular [6,7]. This paper introduces an algorithm for detecting localized virtual concept drift [4] using data sampling density and localized entropy together with a simple method of ensemble construction.

The phenomenon of concept drift arises in systems generating streams of data. Any classifier trained on a currently available sample will become obsolete if the distribution of data and its corresponding labels change. Different classifications were proposed [3] based on the speed (sudden vs gradual) and nature of change (virtual vs real). The "real concept drift" is observed when class labels change reflecting the underlying change to the concept. Most commonly we are dealing with a "virtual" change [5] only to the distribution of data points in the input space.

This paper is structured as follows: first, a measure of information density is introduced, based on a reference distribution of points. Then, an algorithm ("DEnBoost" - Dynamic Environment Boosting) for constructing classifier ensembles is proposed. It uses the change of information density in subsequent steps. Finally, the methodology is validated with experiments on synthetic data.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 164–171, 2011.

2 Definition of the Model

The key observation in the proposed methodology is that most machine learning algorithms model the problem with acceptable accuracy only in the close vicinity of points from the training set. Thus, models trained on small portions of the input space will have high error outside of the known area.

Reference Sampling. Let $D = \{(x_i, t_i) : x_i \in X, t_i \in T\}$ be the training set with input values from X and target labels from the set T. We also put $D_X = \{x_i : (x_i, t_i) \in D\}$ and $D_T = \{t_i : (x_i, t_i) \in D\}$.

Using the *uniform independent and identically distributed sampling* as reference, a notion of *information density* can be given following an observation, that the higher the density of data points and homogeneity of target output values, the better performance is expected on that particular area from a simple classifier. Unbiased min-max estimators can be used to calculate the value interval of a uniform sampling from the same area as the training set:

$$\hat{\zeta}_{\min}(D_X) = \frac{N \min D_X - \max D_X}{N-1} \ , \hat{\zeta}_{\max}(D_X) = \frac{N \max D_X - \min D_X}{N-1} \ , \quad (1)$$

where N is the number of points in training set D. Let $\mathcal{U}_X^N$ be a uniform sampling of N-points from the estimated value interval. The distance between two data points $x, y \in X$ is calculated using any norm $\|x - y\|$, *e.g.* the Euclidean norm $\|x - y\|_2$. In order to describe the reference neighborhood of data points the expected distance to the nearest neighbor is calculated.

Definition 1. *The reference neighborhood radius is taken as the expected distance to the nearest neighbor with the neighborhood of x taken as a ball $\mathcal{B}(x, \alpha r)$, $\alpha \in \mathbb{R}^+$, where*

$$r = E[\| \ x - \underset{y \in \mathcal{U}_X^N \setminus \{x\}}{\operatorname{argmin}} \|x - y\| \ \|], x \in \mathcal{U}_X^N \quad (2)$$

Data and Information Density Measure. Using the reference neighborhood, a measure for data points density and information can be introduced.

Definition 2. *For each point x in the input space X, data density is the number of points from the training set closer to x than αr, where $\alpha \in \mathbb{R}^+$.*

$$\tau(x) = |\{y \in D_X : \|x - y\| \leq \alpha r\}| \quad (3)$$

Data density grows for natural clusters of data points and falls down to 0 in empty areas of the input space.

The calculation of the reference radius requires the expected neighbor distance in $\mathbb{R}^n$. For $\mathbb{R}^2$ this is $\frac{1}{2\sqrt{l}}$, where l is the density of points in one unit of volume. For higher dimensions it can be calculated using a formula by Clark [2] and a later modification by Cherni [1].

Information Density. To form a complete *information density* measure, a *data certainty* measure needs to be chosen first. For classification problems this can be derived from the Shannon Entropy:

Definition 3. *Let D be the training set, r the reference radius, k the number of classes in the training set and $H_k(\xi)$ the Shannon Entropy of ξ. Certainty measure is defined as*

$$\mathcal{C}_r^D(x_i) = \frac{1}{1 + H_k(\{t_j : x_j \in \mathcal{B}(x_i, \alpha r), (x_j, t_j) \in D\})} \ , \tag{4}$$

where $\alpha \in \mathbb{R}^+$.

Definition 4. *Let D denote the training set, r the reference radius, $\tau(x)$ the data density and $\mathcal{C}_r^D(x_i)$ the certainty measure. Then, information density is:*

$$\rho(x) = \tau(x)\mathcal{C}_r^D(x_i). \tag{5}$$

For other radii this becomes $\rho(x, R) = \tau(x)\mathcal{C}_R^D(x_i)$. Information density is largest in areas of tightly packed points with low variance of value.

The DEnBoost Algorithm. Let us now consider a system that generates data in packets. Let D be the initial training set and let D' be the next generated batch. For convenience these could be broken down to D_X, D'_X, D_T and D'_T. The *information density difference* between data sets $D(t)$ and $D(t+k)$ is $\Delta_{D(t),D(t+k)}(x) = \varphi(\rho_i^{D(t+k)}(x) - \rho_i^{D(t)}(x))$, where $\varphi : \mathbb{R} \to \mathbb{R}^+ \cup \{0\}$ is any continuous function fulfilling following requirements:

$$\varphi(0) = 0, \forall y \neq 0 : \varphi(y) \geq 0 \tag{6}$$

$$\forall y > 0, w \geq 0 : \varphi(y + w) \geq \varphi(y) \tag{7}$$

$$\forall y < 0, w \geq 0 : \varphi(y - w) \geq \varphi(y) \tag{8}$$

For the course of this paper, it is assumed that φ is a symmetric function. Density change can be measured in two data sets using a common radius as

$$\Delta^r_{D(t),D(t+k)}(x) = \varphi(\rho_i^{D(t+k)}(x, r) - \rho_i^{D(t)}(x, r)) \ . \tag{9}$$

A new model is built for incoming data packets on a randomly selected subset of all known data points. The sampling probability is equal to the normalized information density change:

$$P(x \in D_{X_2}) = \frac{\Delta^r_{D_X, D_X \cup D'_X}(x)}{\max_{y \in D_X \cup D'_X} \Delta^r_{D_X, D_X \cup D'_X}(y)} \ , \tag{10}$$

where r is the reference radius for the set D_X. At least one point, namely the one with the highest information density change, will enter the new training set. Random selection protects the model from ill-conditioned sampling.

Algorithm 1. init(X,T) – build the first model.

$\mathcal{M}_1$ = buildModel(X,T)
$X_1 = X$; $T_1 = T$; $X_{full} = X$; $T_{full} = T$
$n = 1$

Algorithm 2. add(X',T') – include new data in the composite model.

r = computeReferenceRadius(X_{full})
$X_{temp} = X_{full} \cup X'$; $T_{temp} = T_{full} \cup T'$
W = new array(length(X_{temp}))
for all $x \in X_{temp}$ **do** $W[x] = \Delta^r_{X_{full},X_{temp}}(x)$
W_n = new array(length(X_{temp}))
for all $x \in X_{temp}$ **do** $W_n[x] = (W[x])/(\max W)$
$X_{n+1} = \emptyset$; $T_{n+1} = \emptyset$
for all $x \in X_{temp}$ **do**
    if random($\mathcal{U}(0,1)$) $\leq W_n[x]$ **then** $X_{n+1} = X_{n+1} \cup x$; $T_{n+1} = T_{n+1} \cup T(x)$
$\mathcal{M}_{n+1}$ = buildModel(X_{n+1},T_{n+1}) ; $n = n + 1$
$X_{full} = X_{temp}$; $T_{full} = T_{temp}$

Algorithm 3. simulate(x) – compute the composite model's output.

$W = array(n)$
for $i = 1$ to n **do**
    R = computeReferenceRadius(X_i)
    $W[i] = \rho_i^{X_i}(x, \gamma R)$
return $(\sum_{i=1}^{n} W[i] * sim(\mathcal{M}_i, x))/(\sum_{j=1}^{n} W[j])$

Submodels are combined with a simple weighted output average, with each training subset's information density used as a weight:

$$\mathcal{M}(x) = \frac{\sum_{i=1}^{N} \rho^{D_{X_i}}(x, \gamma r_i)\mathcal{M}_i(x)}{\sum_{i=1}^{N} \rho^{D_{X_i}}(x, \gamma r_i)} \quad , \tag{11}$$

where $\gamma \in \mathbb{R}^+$ is called a *generalization factor*, $\mathcal{M}_i(x)$ is the output of the i-th submodel and r_i is the reference radius for D_{X_i}. Smaller values of γ limit the influence of submodels on the composite output, while larger encourage influences between data points. In the areas with 0 information density value, the output of the first submodel is used.

The Continuous DEnBoost Algorithm. The standard DEnBoost algorithm has a number of issues: ensemble corrections are confined to areas limited by the reference radius, crisp boundaries are formed between areas with different information density and classification in the areas with 0 information density depends on the accuracy of the first subclassifier. In order to fix these issues, smooth measures are derived using radial basis functions centered at observations.

Continuous Information Density. Let $\mathcal{N}_{\mu,\sigma}(x)$ denote the probability density function of the normal distribution, and $\mathcal{N}^n_{\mu,\sigma}(x)$ a normalized version, *i.e.* $\mathcal{N}^n_{\mu,\sigma}(x) = (\mathcal{N}_{\mu,\sigma}(x))/(\mathcal{N}_{\mu,\sigma}(0))$.

Definition 5. *Let $x \in X$ and r be the reference radius for D_X. The continuous data density is given by:*

$$\tau^c(x) = \sum_{y \in D_X} \mathcal{N}^n_{0,r}(||x - y||) \ . \tag{12}$$

Note, that the reference radius is used as the standard deviation parameter for the Gaussian distribution. For convenience, a similar function $\tau^c_k(x)$ takes into account only points from class k allowing to define a soft class probability density as $p_y(x_i) = \tau^c_{x_i}(y)/\tau^c(y)$. This is used to form a modified, distance-weighted Shannon's Entropy and continuous information density:

Definition 6. *Let X denote a random variable with possible values $\{x_1, \ldots, x_n\}$ and $x \in X$. The Radial Basis Function Weighted Entropy is defined as:*

$$H^{RBF}_b(x) = -\sum_{i=1}^{n} p_x(x_i) \log_b p_x(x_i) \ . \tag{13}$$

Definition 7. *Let D denote the training set, r the reference radius and k the number of classes in the training set. The continuous certainty measure is defined as $\mathfrak{C}^D_r(x_i) = 1/(1 + H^{RBF}_k(x_i))$.*

Definition 8. *Let D denote the complete training, r the reference radius, $\tau^c(x)$ the continuous data density and $\mathfrak{C}^D_r(x_i)$ the continuous certainty measure. Then the continuous information density is defined as $\rho^c(x, r) = \tau^c(x)\mathfrak{C}^D_r(x_i)$.*

The difference in information density measured by the standard metric and by the continuous one is visible in the Fig. 1.

3 Experiments

At the current development stage, a comparison of DEnBoost with other algorithms would be misleading. Therefore, two experiments on synthetic data have been conducted. A data generating system was simulated with a random mixture of Gaussian in $\mathbb{R}^2$. Clouds were placed in the input space with their positions and radii drawn from a uniform distribution. Each cloud was given a label and a weight. Observations were generated by selecting a cloud based on its weight and randomizing a point based on cloud's parameters. The initial data batch consisted of 40 observations and in each step, a new batch with 20 observations was generated. Depending on the experiment, cloud weights were changed to simulate virtual concept drift. The final test-sample consisted of 200 observations with cloud weights set to 1. For each experiment and parameter combination, 70

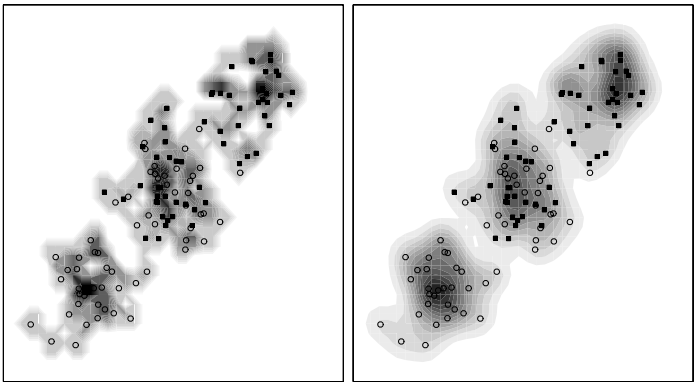


Fig. 1. Information density measured with the standard measure (left) and the continuous measure (right). Note the increased smoothness of the continuous measure.

Table 1. Mean value (μ) and standard deviation (σ) of accuracy for the first subclassifier and the ensemble in the no-drift experiment (top) and the virtual concept drift experiment (bottom) with both information density measures

Inf. dens.	Classes	Clouds	First μ	First σ	Ens. μ	Ens. σ	μ Change	σ Change
Stand.	2	3	0,916	0,094	0,924	0,081	**0,008**	**0,061**
Stand.	2	5	0,852	0,113	0,883	0,087	**0,030**	**0,072**
Stand.	2	7	0,810	0,100	0,839	0,081	**0,028**	**0,074**
Stand.	4	5	0,698	0,077	0,822	0,105	**0,123**	**0,099**
Stand.	4	7	0,607	0,105	0,720	0,100	**0,112**	**0,104**
Stand.	6	7	0,481	0,070	0,655	0,108	**0,174**	**0,108**
Cont.	2	3	0,925	0,113	0,933	0,088	**0,008**	**0,072**
Cont	2	5	0,855	0,131	0,881	0,108	**0,026**	**0,071**
Cont	2	7	0,836	0,115	0,864	0,088	**0,028**	**0,062**
Cont	4	5	0,711	0,057	0,801	0,104	**0,089**	**0,094**
Cont	4	7	0,588	0,101	0,734	0,111	**0,146**	**0,109**
Cont	6	7	0,477	0,066	0,659	0,074	**0,181**	**0,092**

Inf. dens.	Classes	Clouds	First μ	First σ	Ens. μ	Ens. σ	μ Change	σ Change
Stand.	2	3	0,767	0,173	0,936	0,083	**0,168**	**0,141**
Stand.	2	5	0,781	0,126	0,883	0,084	**0,101**	**0,111**
Stand.	2	7	0,772	0,112	0,837	0,102	**0,065**	**0,064**
Stand.	4	5	0,603	0,116	0,808	0,098	**0,204**	**0,114**
Stand.	4	7	0,629	0,106	0,830	0,095	**0,200**	**0,123**
Stand.	6	7	0,455	0,075	0,672	0,097	**0,217**	**0,110**
Cont	2	3	0,829	0,162	0,953	0,068	**0,124**	**0,144**
Cont	2	5	0,799	0,120	0,882	0,082	**0,082**	**0,107**
Cont	2	7	0,805	0,142	0,883	0,097	**0,078**	**0,100**
Cont	4	5	0,604	0,114	0,786	0,105	**0,181**	**0,113**
Cont	4	7	0,608	0,085	0,735	0,101	**0,127**	**0,091**
Cont	6	7	0,427	0,068	0,657	0,089	**0,229**	**0,092**

runs were conducted, each with 4 incremental steps. Subclassifiers were chosen to be simple feed-forward neural networks with 2 hidden neurons (larger nets had no impact on the performance).

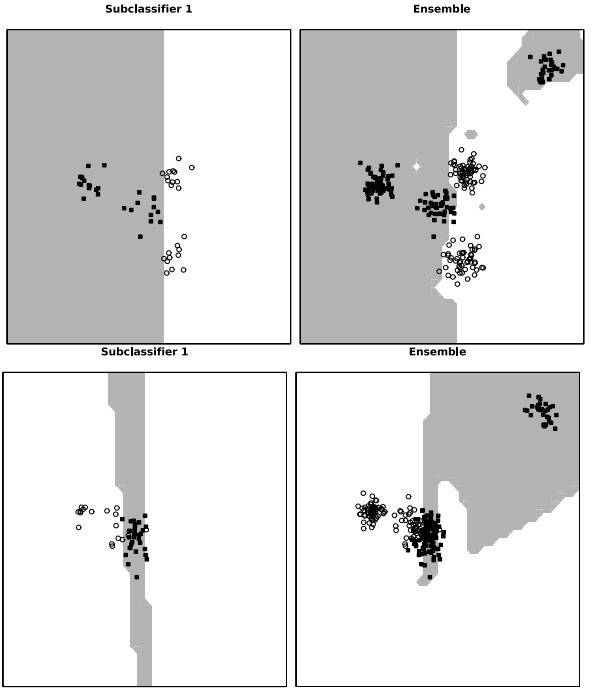


Fig. 2. First classifier (on the left) and an ensemble (on the right) constructed using standard (top row) and continuous (bottom row) information density measure. Note, that the algorithm with the standard measure adjusts classification only in the neighborhood of data points, while the continuous version allows generalization between observations.

In the first experiment, cloud weights did not change between steps. This tested whether the algorithm retains the performance when no virtual concept drift is present in the system. The experiment confirmed that the proposed architecture can introduce minor adjustments to classification boundaries (see Tab. 1).

In the second experiment one of the clouds was selected and had its weight interpolated from 0 to 1 between steps, *i.e.* no points from the selected cloud were present in the initial training set, while more and more appeared throughout the run. The initial classifier was augmented using the DEnBoost algorithm to produce correct classifications. A significant increase in the accuracy was observed indicating that the algorithm has introduce new knowledge into the ensemble. No difference in performance was observed between information density versions, but an inspection of classification boundaries revealed good generalization property of the continuous measure (see Fig. 2). In the proposed experiments, statistical comparison of accuracies is unnecessary. The goal is to provide evidence that the algorithm increases performance of the base classifier and does not degrade the accuracy on already known areas of the input space.

4 Discussion

The DEnBoost algorithm is successful at solving the virtual concept drift problem, because information density measure allows it to detect locally novel information. The distribution of observations and their corresponding labels defines focus areas for subsequent classifiers and ensemble members are combined using their local competence. The continuous information density provides a good generalization far from known observations. Furthermore, the algorithm does not depend on a chosen classifier architecture and does not require much parameter adjustments.

Initial results for the methodology are very promising. At its current form, the DEnBoost algorithm is fully capable of increasing model accuracy. Future work will focus on using information density measure together with a more robust model composition strategy to solve the real virtual concept drift problem.

References

1. Cherni, S.: Nearest Neighbor Method, http://www.mcs.sdsmt.edu/rwjohnso/html/sofiya.pdf
2. Clark, P.J., Evans, F.C.: Distance to Nearest Neighbor as a Measure of Spatial Relationships in Populations. Ecology 35(4), 445–453 (1954)
3. Kuncheva, L.: Classifier Ensembles for Changing Environments. In: APWeb 2004. LNCS, vol. 3007, pp. 1–15. Springer, Heidelberg (2004)
4. Tsymbal, A.: The Problem of Concept Drift: Definitions and Related Work. Technical report, Trinity College Dublin (2004)
5. Widmer, G., Kubat, M.: Effective Learning in Dynamic Environments by Explicit Context Tracking. In: Brazdil, P.B. (ed.) ECML 1993. LNCS, vol. 667, pp. 227–243. Springer, Heidelberg (1993)
6. Muhlbaier, M.D., Polikar, R.: An Ensemble Approach for Incremental Learning in Nonstationary Environments. In: Haindl, M., Kittler, J., Roli, F. (eds.) MCS 2007. LNCS, vol. 4472, pp. 490–500. Springer, Heidelberg (2007)
7. Susnjak, T., Barczak, A., Hawick, K.: Adaptive Ensemble Based Learning in Nonstationary Environments with Variable Concept Drift. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010, Part I. LNCS, vol. 6443, pp. 438–445. Springer, Heidelberg (2010)

On Performance of DRSA-ANN Classifier

Urszula Stańczyk

Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland

Abstract. Rule-based and connectionist classifiers are typically named as two different approaches to recognition tasks. The first relies on induction of a set of rules that list conditions to be met for a decision to be applicable, while the latter means distribution of data and processing. Both solutions give satisfactory results in many classification problems yet their fusion and analysis of performance of the resulting hybrid classifier bring additional observations as to the role of particular features in the recognition. These observations are not based on domain knowledge, but on techniques employed and their inherent properties. The paper presents a study on performance of DRSA-ANN classifier applied within the domain of stylometry, a quantitative analysis of writing styles.

Keywords: DRSA, ANN, Classifier, Feature Selection, Stylometry.

1 Introduction

Hybrid artificial intelligence systems are popularly used in pattern recognition due to their flexibility that enables adaptation to particularities of studied problems. A fusion of classifiers is usually attempted in order to enhance quality, to increase the correct recognition ratio. It leads to finding such solutions that fit better with respect to the considered criteria [10].

A fusion of processing techniques can be treated as a goal in itself, but also as a means to view some problem from a different perspective. This could be most informative as all approaches possess their own inherent properties, particularly useful when they lead to reduction of dimensionality [3].

Dominance-based Rough Set Approach (DRSA) is a rule-based solution to classification problems. This methodology bases on fundamental concepts as defined by Z. Pawlak, but substitutes the indiscernibility relation of the classical version (CRSA) with the dominance, which allows to recognise not only the presence or absence of some properties, but also their ordering which leads to dealing with not just abstract or discrete, but continuous numeric data [4,6].

In DRSA processing some features can be considered as more important than others, some can be disregarded if a relative reduct (a subset of attributes still maintaining correct classification) is selected for further processing. When decision rules are calculated they are characterised, among others, by their length and support - the number of learning samples they are valid for. All these elements are taken under consideration when constructing DRSA classifier.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 172–179, 2011.

Distributed data and processing in artificial neural networks constitute the connectionists approach within which the importance of features is found in the learning process. A network first assumes then modifies weights associated with interconnections in such a way that leads to arriving at the structure that returns the minimal error at the network output.

Past research shows that rough set-based analysis can be exploited to reduce the number of inputs for ANN classifier by observing the number of occurrences of conditional attributes in relative reducts for discretised and real data, and in decision rules for real data [7,8]. The paper is a continuation of this research track and it concentrates on supports of decision rules, which is considered as an importance indicator for individual attributes. The performance of the hybrid classifier is studied in the area of stylometric analysis of texts, which assumes that any writing style can be uniquely defined by quantitative measures [1].

2 Stylometric Analysis of Texts and Input Data

Stylometry is such a branch of textual analysis that enables expressing individual writing styles by means of quantitative measures. It allows to find unique characteristics of authors, detect some traits that they share, establish authorship [2]. In the area of forensic applications it is employed to discover plagiarism, to verify ransom notes, to confirm identity of terrorist threat-makers. Historic and literary applications include proving authenticity of written works.

As it is much easier nowadays to imitate somebody else's work, due to the commands of copy and paste of contemporary word processors, finding textual markers to be used in the analysis is not simple. That is why, instead of exploiting some striking language features that are easily spotted, stylometry observes more subtle patterns of writing, reflecting on commonly but rather subconsciously employed function words or punctuation marks [5].

As the input data for all experiments there were taken literary works of two writers: Thomas Hardy and Henry James. All samples consisted of characteristics calculated for fragments (of approximately the same length, a chapter if possible) coming from novels. There were 180 learning and 80 testing samples.

To recognise authorship the base set of features comprised 25 elements and they were frequencies of usage for lexical and syntactic markers: but, and, not, in, with, on, at, of, this, as, that, what, from, by, for, to, if, a fullstop, a comma, a question mark, an exclamation mark, a semicolon, a colon, a bracket, a hyphen.

3 Construction of a Connectionist Classifier

As a general structure for an artificial neural network used in tests there was chosen Multilayer Perceptron with sigmoid activation function and Backpropagation algorithm as the learning rule [9]. The number of ANN input neurons corresponded to the number of considered characteristic features so it started with 25, then was decreased. The number of outputs was equal to the number of classes, corresponding to two recognised authors.

To establish the internal structure of ANN-classifier with the highest classification accuracy for the task, firstly there were tested networks with varying numbers of hidden layers and neurons. Out of all trials finally for all other experiments there was selected the structure with two hidden layers with the total number of neurons in them equal to the number of inputs, the first layer containing $\lceil 3/4$ number of inputs$\rceil$ neurons, the second $\lfloor 1/4$ number of inputs$\rfloor$.

To minimise the influence of the initiation of weights on a network classification accuracy, there was used multi-starting approach: for each network configuration the training was performed 20 times. Basing on such series there was obtained the average network performance and only these results are presented in the paper. For the base set of 25 characteristic features the average network classification accuracy was 82.5%.

There exist plenty of approaches and methods for feature selection and reduction, dedicated for ANN or other classifiers, each with some advantages and disadvantages [3]. The quality of obtained results is always to some degree task-dependent and comparisons to some alternative solutions practically endless. In the research described feature reduction is employed not as a goal in itself but rather as a means to observe the importance of individual characteristic features. Therefore, the paper limits considerations to the performance of the connectionist classifier with selection of features based on rough analysis only.

4 Analysis of DRSA Decision Rules

Within the first step of DRSA methodology there is constructed a decision table [4], containing available knowledge about all objects of the universe. This table was based on the set of training samples that was prepared for the connectionist classifier used in research. The columns of conditional attributes corresponded to the network inputs, while the values of the single decision attribute were the same as the expected classes of recognised objects. Next there were calculated decision rules, specifying conditions to be met for any decision to be applicable.

The *premise* part of a decision rule is a conjunction of elementary conditions, which concern individual attributes, while the *decision* part of the rule indicates an assignment to a decision class or union of classes. The rules are induced in order to cover objects from lower approximations of sets being decision classes or their unions and can be certain, possible or approximated. The rules are found through either exhaustive search for all, to provide only the minimal cover of the learning samples, or to satisfy some additional criteria.

Each rule is supported by some number of learning examples and this support can be treated as an indicator of the rule importance. This importance reflects on individual attributes the rules refer to. Within the total number of 46,191 decision rules found, their supports ranged from 1 to 64, as shown in Table 1, along with the ordering of attributes used in the previous tests. There are relatively few rules with support values in the highest range, while the majority falls in the lowest range of values.

Some conditional attributes are used more often than others in construction of rules, which was used in the past research to show how this frequency of usage

Table 1. Ordering of attributes based on the number of decision rules regardless of their support

Number	Support in range								
of rules	1-10	11-20	21-30	31-40	41-50	51-60	61-64	Attribute	Order
3928	3872	42	12	2	0	0	0	but	
4173	3256	530	214	95	56	21	1	and	L1
6167	6117	39	6	3	1	1	0	that	L2
6173	6069	91	10	3	0	0	0	what	M7
7450	7370	74	5	1	0	0	0	for	L3
7469	7355	95	12	4	3	0	0	?	
7615	6651	669	216	51	25	3	0	from	
7692	7545	111	30	3	3	0	0	if	
7951	7589	270	54	18	14	6	0	(	
7997	7845	127	20	5	0	0	0	-	M6
8451	7575	639	163	48	15	11	0	by	L4
8472	8279	164	26	2	1	0	0	as	
8647	8570	55	16	2	2	2	0	with	M5
9083	8953	109	16	5	0	0	0	at	L5
9798	9683	93	17	5	0	0	0	;	M4
10241	9720	376	105	28	10	2	0	in	L6
10306	9736	444	92	24	4	5	1	not	
10327	9928	325	59	12	3	0	0	:	
10640	10307	260	52	18	3	0	0	!	M3
11005	10795	176	29	5	0	0	0	.	L7
11177	10913	232	26	5	1	0	0	,	
11427	11177	188	42	11	8	1	0	this	
11839	11579	206	40	8	3	3	0	to	M2
12922	12523	324	57	17	1	0	0	on	M1
13311	12921	303	66	15	6	0	0	of	

influences the power of ANN classifier [8]. Following this track of reasoning some attributes appear more often in rules with lower support, while some with higher.

5 Obtained Results

When support of rules is taken into account it is necessary to consider what is more important: a single rule with very high support or several rules with some lower, yet not minimal support. As the answer to such question is not necessarily straightforward it is best to verify it with tests. This in turn leads to several possible orderings of features. The first one, presented in Table 2, is based on percentage of rules that are in at least median range of support.

As there are only some few rules with high support, it can be argued that they reflect some specific patterns present in the learning set which may not necessarily be detectable in the testing set. It is possible that playing it safe and taking into account the median support range will bring better results. This gives the second ordering, shown in Table 3a.

Table 2. Ordering of attributes based on the number of rules with support in median and higher ranges (Order 1)

Nr of rules with supp. in range [%]					
31-40	41-50	51-60	61-64	Attribute	Order 1
0.013	0.000	0.000	0.000	for	
0.045	0.000	0.000	0.000	.	L1
0.048	0.000	0.000	0.000	what	
0.050	0.000	0.000	0.000	but	
0.051	0.000	0.000	0.000	;	
0.055	0.000	0.000	0.000	at	M12
0.062	0.000	0.000	0.000	-	L2 M11
0.131	0.007	0.000	0.000	on	L3
0.044	0.008	0.000	0.000	,	
0.023	0.011	0.000	0.000	as	M10
0.169	0.028	0.000	0.000	!	L4
0.116	0.029	0.000	0.000	:	M9
0.039	0.039	0.000	0.000	if	L5
0.053	0.040	0.000	0.000	?	
0.112	0.045	0.000	0.000	of	M8
0.096	0.070	0.008	0.000	this	L6 M7
0.048	0.016	0.016	0.000	that	L7 M6
0.273	0.097	0.019	0.000	in	L8
0.023	0.023	0.023	0.000	with	
0.067	0.025	0.025	0.000	to	M5
0.669	0.328	0.039	0.000	from	L9 M4
0.226	0.176	0.075	0.000	(	M3
0.567	0.177	0.130	0.000	by	M2
0.232	0.038	0.048	0.009	not	M1
2.276	1.341	0.503	0.023	and	

The third order (Table 3b) relies on intuitive thinking that what is good for one classification methodology should also bring good results for the other. That is why there is considered the maximal support of rules for all attributes.

When comparing these orderings it is clear that in some parts they are similar to each other, which brings the question if really all should be tried. The answer to that and the motivation for performed tests was that the point of interest is to observe trends in the classifier performance while following various orderings of attributes, to study how different focus on a rule support reflects on attributes and through that on classification accuracy of ANN.

Following the three orderings of attributes, there were conducted three groups of tests, each further divided into two parts: one with keeping the attributes considered as the most important in the current perspective with disregarding the least (denoted by "L"), second with removing the most important while keeping the least important features (denoted by "M"). For comparison in the following graphs there are also given results from the past tests where just the numbers of rules were considered, regardless of their support.

Table 3. Rule support-based ordering of attributes: a) Order 2 - by number of rules with support in median range, b) Order 3 - by maximal support of rules

a)

Rules with support in 31-40 range [%]	Attribute	Order 2	
0.013	for		
0.023	with	L1	
0.023	as		
0.039	if	L2	
0.044	,	L3	
0.045	.		
0.048	what		
0.048	that		
0.050	but		
0.051	;		
0.053	?		
0.055	at		M7
0.062	-	L4	
0.067	to		M6
0.096	this	L5	M5
0.112	of	L6	
0.116	:		M4
0.131	on	L7	
0.169	!		M3
0.226	(	L8	
0.232	not		
0.273	in		M2
0.567	by		
0.669	from		M1
2.276	and		

b)

Maximal support	Attribute	Order 3	
34	for		
36	at	L1	
37	but	L2	
38	.	L3	
38	-		
39	;	L4	
40	what	L5	
41	as	L6	M11
43	on	L7	M10
45	,	L8	
45	!		M9
46	?	L9	M8
49	of	L10	
49	:		M7
50	if	L11	M6
52	this	L12	M5
56	in	L13	M4
57	to	L14	M3
58	that		M2
59	with		
59	from		
59	by		
59	(		M1
64	and		
64	not		

Fig. 1a shows that when ordering with respect to support of rules in median and higher ranges of values, there is maintained the same or even increased classification ratio when keeping the most and discarding the least important attributes even when there are as few as 9 network inputs left. The opposite approach, with reducing the most important features and keeping these less important, quickly causes worsened performance.

For Order 2, with respect to the rules with support in median range (Fig. 1b), the results are not as good, but again the best are obtained while removing less important features. The reverse brings visible decrease in the network performance, even if not as dramatic as when considering just the numbers of rules.

The last group of tests focused on the maximal support of rules for all attributes (Fig. 2). When discarding the least while keeping the most important attributes the networks maintained classification accuracy at the same or higher level as for the whole set of attributes even when there were only 8 inputs left.

Within these subsets of attributes considered neither one is reduced in such a way that gives only lexical or only syntactic markers. Instead, they are always

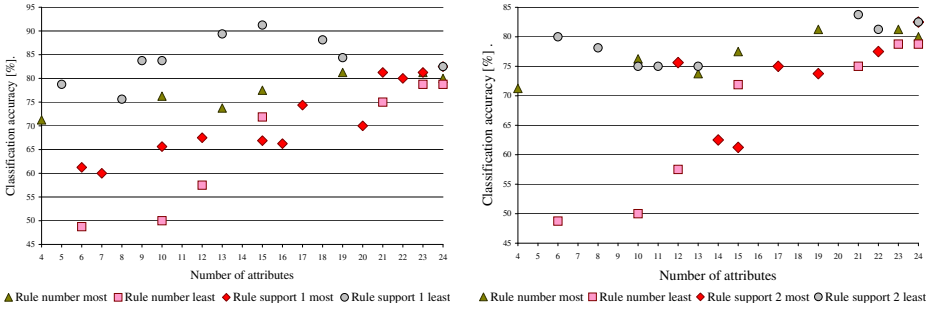


Fig. 1. ANN classification accuracy in relation to the number of attributes with feature reduction based on: a) Order 1 (number of rules with supports in at least median range), b) Order 2 (number of rules with supports in median range)

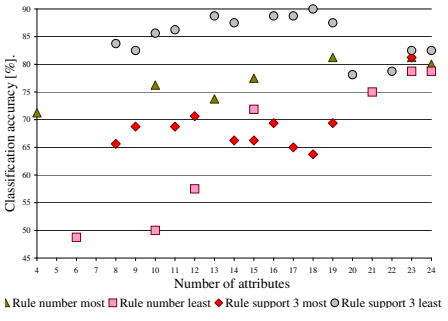


Fig. 2. ANN classification accuracy in relation to the number of attributes with feature reduction based on Order 3 of attributes (maximal supports of rules)

mixed. For all three orderings there are several subsets that bring some increase in classification accuracy while at the same time offering noticeable feature reduction, and in these subsets there are some re-occurring attributes. Not in the same order, but among more significant features there are always listed four lexical descriptors that reflect the usage of "and", "not", "by", "from", and a syntactic marker for a bracket. The base five markers can be considered as essential for correct classification in this particular task of authorship attribution. Removing these essential features causes worsening performance of the classifier even when there are still several attributes left.

6 Conclusions

The paper presents research on importance of characteristic features for a connectionist classifier as seen from a rule-based approach perspective. Within DRSA methodology the importance of decision rules can be perceived by the number

of learning samples they are valid for. Analysis of attributes with respect to supports of rules they form leads to their ordering that is next imposed on ANN classifier. Experiments show that the best results are obtained when observing the maximal supports of rules for all attributes. When keeping these attributes present in rules with the highest support the number of ANN inputs can be reduced to less than 33% of the initial set while maintaining the same classification accuracy. These results indicate that the importance of attributes as perceived in DRSA methodology can be used to advantage for ANN classifier. Such perspective requires no additional domain knowledge about features and leaves the task of discovering the importance of attributes to processing techniques employed.

Acknowledgments. 4eMka Software used in search for decision rules [4,6] is available at a website of Laboratory of Intelligent Decision Support Systems, Poznan University of Technology (`http://www-idss.cs.put.poznan.pl/`), Poland.

References

1. Burrows, J.: Textual analysis. In: Schreibman, S., Siemens, R., Unsworth, J. (eds.) A companion to digital humanities, ch. 23, Blackwell, Oxford (2004)
2. Craig, H.: Stylistic analysis and authorship studies. In: Schreibman, S., Siemens, R., Unsworth, J. (eds.) A companion to digital humanities, Blackwell, Oxford (2004)
3. 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)
4. Greco, S., Matarazzo, B., Slowinski, R.: Dominance-based rough set approach as a proper way of handling graduality in rough set theory. Transactions on Rough Sets 7, 36–52 (2007)
5. Peng, R.D., Hengartner, H.: Quantitative analysis of literary styles. The American Statistician 56(3), 15–38 (2002)
6. Słowiński, R., Greco, S., Matarazzo, B.: Dominance-based rough set approach to reasoning about ordinal data. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 5–11. Springer, Heidelberg (2007)
7. Stańczyk, U.: Relative reduct-based selection of features for ANN classifier. In: Cyran, K.A., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. Advances in Intelligent and Soft Computing, vol. 59, pp. 335–344. Springer, Heidelberg (2009)
8. Stańczyk, U.: Rough set-based analysis of characteristic features for ANN classifier. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 565–572. Springer, Heidelberg (2010)
9. Waugh, S., Adams, A., Twedeedie, F.: Computational stylistics using artificial neural networks. Literary and Linguistic Computing 15(2), 187–198 (2000)
10. 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)

Performance Analysis of Fuzzy Aggregation Operations for Combining Classifiers for Natural Textures in Images

María Guijarro, Gonzalo Pajares, P. Javier Herrera, and J.M. de la Cruz

Dpt. Ingeniería del Software e Inteligencia Artificial, Facultad Informática, Universidad Complutense, 28040 Madrid, Spain
{mguijarro,pajares}@fdi.ucm.es, pjherrera@pdi.ucm.es, jmcruz@fis.ucm.es

Abstract. One objective for classifying pixels belonging to specific textures in natural images is to achieve the best performance in classification as possible. We propose a new unsupervised hybrid classifier. The base classifiers for hybridization are the Fuzzy Clustering and the parametric Bayesian, both supervised and selected by their well-tested performance, as reported in the literature. During the training phase we estimate the parameters of each classifier. During the decision phase we apply fuzzy aggregation operators for making the hybridization. The design of the unsupervised classifier from supervised base classifiers and the automatic computation of the final decision with fuzzy aggregation operations, make the main contributions of this paper.

Keywords: classifier combination, fuzzy aggregation, parametric estimation, fuzzy clustering, Bayes classifier.

1 Introduction

Nowadays the technology demands solutions for various applications. The classification of individual pixels belonging to natural textures is one of such applications due to the high spatial resolutions achieved in the images. The areas where textures are suitable include agricultural crop ordination, forest or urban identifications and damages evaluation in catastrophes or dynamic path planning during rescue missions or intervention services also in catastrophes (fires, floods).

In this work we use a pixel-based approach under the RGB color space representation because, as reported in [1], it performs favorably against other color mappings. Hence, the three RGB spectral values are the features used in our method. The same texture could be displayed with different RGB levels; this makes the problem fuzzy in nature, justifying the choice of the fuzzy aggregation operation.

Nevertheless, the main problem is: which is the best strategy for combining simple classifiers? This is an issue still open. Here we find one of the problems solved with techniques based on hybrid artificial intelligence systems [12]. Indeed in [3] it is stated that the best combination method does not exist. In [5] a revision of different approaches is reported including the way in which the classifiers are combined. Some important conclusions are: 1) if only labels are available, a majority vote should be suitable; 2) if continuous outputs like posterior probabilities are supplied, an average

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 180–188, 2011.

or some other linear combinations are suggested; 3) if the classifier outputs are interpreted as fuzzy membership values, fuzzy approaches could be used; 4) also it is possible to train the output classifier separately using the outputs of the input classifiers as new features. In 1) a selection criterion is applied, in 2 and 3) a fusion strategy is carried out and in 4) a hierarchical approach is used [4,10].

Because we have available continuous outputs, we propose a new fusion approach which combines two base classifiers: the fuzzy clustering (FC) and the probabilistic parametric Bayesian (BP) approach [7]. The following two phases are involved: training and decision. These classifiers are selected because they provide the best performance when used in a subset of images which are to be classified. Moreover they have been broadly applied in the literature with high performances. Both FC and BP estimate their parameters which are stored in the Knowledge Base (*KB*). During the classification or decision phase, each base classifier provides for each pixel a support of belonging to a cluster (FC gives membership degrees and BP probabilities). We propose the hybrid approach where the individual supports are combined through the fuzzy aggregation operations. The results are better than those obtained by the simple classifiers, as the non-parametric approach Parzen´s window or the vector quantization [7]. This combined scheme, joined to the design of the unsupervised strategy, makes the main contribution to the hybrid systems.

2 Automatic Hybrid Classifier Design

Our system works in two stages: 1) performing a training process with a set of patterns; 2) performing a classification process, the final decision is the class that the new patterns are classified as belonging to. Figure 1 shows both processes.

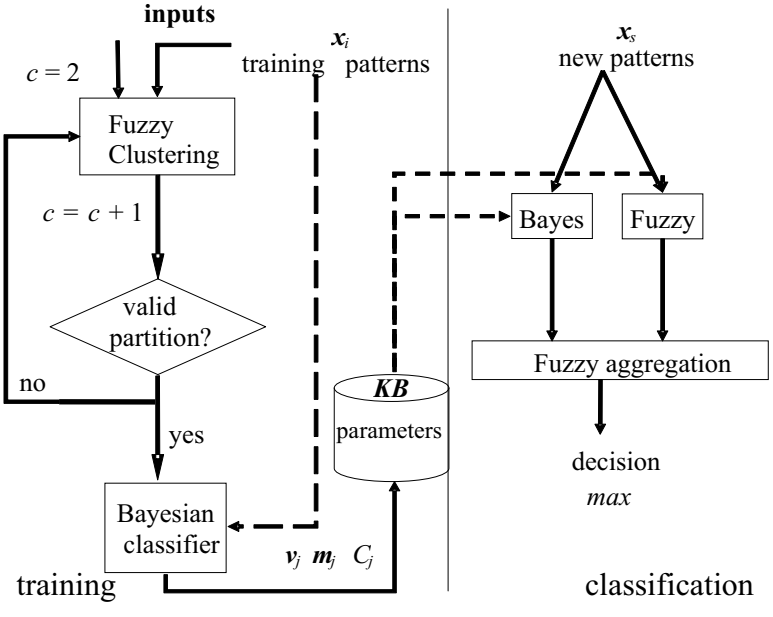


Fig. 1. Architecture of the hybrid classifier based on the fuzzy aggregation approach

In both processes, training and classification, each pattern is characterized by a feature vector $\boldsymbol{x}$. As mentioned before, in this paper, we use a pixel-based approach and taking into account that we are classifying multiespectral textured images, we use as the attribute vector the spectral components, i.e. the red, green and blue. The RGB map performs better than other colour representations [2]. So, $\boldsymbol{x}$ is a 3-dimensional vector representing each pixel, where its components are the red, green and blue values respectively.

A. The Training Process

During the training phase, we start with the observation of a set X of n training patterns, i.e. $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\} \in \Re^3$. Each sample is to be assigned to a given class, where the number of possible classes is c. Each class is identified as w_j, where $j = 1 \ldots c$. Now, the problem is to assign each pattern sample to a class and compute the cluster prototypes. For such purpose we have chosen the well-tested fuzzy clustering framework which has been customized and tailored for working in an unsupervised fashion according to the criterion described in [7]. The original FC computes for each $\boldsymbol{x}_i$ at the iteration k its membership grade and updates the cluster centers according to equation (1),

$$\mu_{ij}(k) = \frac{1}{\sum_{r=1}^{c} \left(d_{ij}(k) / d_{ir}(k) \right)^{2/(m-1)}} \qquad \boldsymbol{v}_j(k+1) = \frac{\sum_{i=1}^{n} \mu_{ij}^m(k) \boldsymbol{x}_i}{\sum_{i=1}^{n} \mu_{ij}^m(k)} \tag{1}$$

d is the squared Euclidean distance, m is called the exponent weight, $\boldsymbol{v}$ are the cluster centers, and μ are the membership grade. The stopping criterion of the iteration process is achieved when $\left\| \mu_{ij}(k+1) - \mu_{ij}(k) \right\| < \varepsilon \ \ \forall ij$ or a number N of iterations is reached. The number of classes is initially set to 2. After the fuzzy clustering process, a partition of the input training patterns is obtained, where each cluster j has associated its center $\boldsymbol{v}_j$. Also, for each sample i its corresponding membership grades, μ_{ij}, of belonging to each cluster j, are computed. The cluster centres are stored in the *KB* to be recovered later.

The next step consists of the cluster validation. This is carried out by computing the *partition coefficient* for the number of classes specified as follows

$$PC(U;c) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \left(\mu_{ij} \right)^2 \tag{2}$$

The maximum value of *PC* for different values of c determines the best partition, i.e. the best number of classes for the set of training samples available. Values of *PC* near the unity indicate that the partition is acceptable. This is because the *PC* is upper bounded by the unity.

Following the scheme in figure 1, the partition of clusters is transferred to the Bayesian classifier [7]. Under this framework the problem is reduced to compute the probability of belonging to w_j given a sample $\boldsymbol{x}$. This parametric classifier receives the validated partition, supplied by FC and estimates the cluster centres $\boldsymbol{m}_j$ and the

covariance matrices C_j as parameters to be stored in *KB* and recovered later. A common practice is to express the likelihood as a Gaussian function of the estimated parameters.

$$p\left(\boldsymbol{x} \mid w_j\right) = \frac{1}{\left(2\pi\right)^{d/2}\left|C_j\right|^{1/2}} exp\left[-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{m}_j\right)^t C_j^{-1}\left(\boldsymbol{x}-\boldsymbol{m}_j\right)\right]$$
$$\boldsymbol{m}_j = \frac{1}{n_j}\sum_{k=1}^{n_j}\boldsymbol{x}_k \;;\; C_j = \frac{1}{n_j-1}\sum_{k=1}^{n_j}\left(\boldsymbol{x}_k-\boldsymbol{m}_j\right)\left(\boldsymbol{x}_k-\boldsymbol{m}_j\right)^t \tag{3}$$

B. The Classification Process: Fuzzy Aggregation Framework as a Combiner

During the classification process new images, and consequently new texture patterns, are to be processed by the system. With such purpose, we recover the $\boldsymbol{v}_j$ cluster centers, the covariance matrices C_j and the $\boldsymbol{m}_j$ mean clusters, which were stored in *KB* during the training process. The original Bayes classifier includes the a priori probability to be combined with the likelihood in the computation of the a posterior probability, but given a pixel to be classified we do not know nothing about its assignment to the clusters, therefore we must assume that a priori, before the observation, all pixels have identical a priori probabilities. Therefore the a priori probability is not discriminant, it can be avoided and the decision can be made only based on the likelihoods estimated according to equation (3).

The fuzzy logic framework provides a number of functions for aggregating two or more fuzzy sets. In this paper, we will introduce these functions and give details about how to use them during classification phase.

The combination is carried out taking into account the supports provided by the selected classifiers, *FC* and *BP*. *FC* and *BP* provide respectively as supports the membership degree $\mu_j(\boldsymbol{x})$ and probability $p_j(\boldsymbol{x})$ that a pattern $\boldsymbol{x}$ belongs to a cluster c_j. Assuming that $p_j(\boldsymbol{x})$ is a fuzzy magnitude ranging in [0, 1] we combine both supports through the aggregation operations defined below. Given the pattern sample $\boldsymbol{x}$ with the supports provided by the simple classifiers $\mu_j(\boldsymbol{x})$ and $p_j(\boldsymbol{x})$ the combination is carried out based on the fuzzy aggregation operations from (4) to (15) [8,9]. After experimentation, in this paper the γ parameters (equations HI and HU) and r (equations YI and YU) are set to *2,* and α (equations DPI and DPU) and γ (equations WI and WU) are set to *0.5*.

Hamacher Intersection (HI) and Hamacher Union (HU):

$$d_j(\boldsymbol{x}) = \frac{\mu_j(\boldsymbol{x})\,p_j(\boldsymbol{x})}{\gamma+(1-\gamma)\left(\mu_j(\boldsymbol{x})+p_j(\boldsymbol{x})-\mu_j(\boldsymbol{x})\,p_j(\boldsymbol{x})\right)} \tag{4}$$

$$d_j(\boldsymbol{x}) = \frac{(\gamma-1)\mu_j(\boldsymbol{x})\,p_j(\boldsymbol{x})+\mu_j(\boldsymbol{x})+p_j(\boldsymbol{x})}{1+\gamma\mu_j(\boldsymbol{x})\,p_j(\boldsymbol{x})} \tag{5}$$

Yager Intersection (YI) and Yager Union (YU):

$$d_j(\boldsymbol{x}) = 1 - \min\left\{1, \left(\left(1-\mu_j(\boldsymbol{x})\right)^r + \left(1-p_j(\boldsymbol{x})\right)^r\right)^{\frac{1}{r}}\right\} \tag{6}$$

$$d_j(\boldsymbol{x}) = \min\left\{1, \left(\left(\mu_j(\boldsymbol{x})\right)^r + \left(p_j(\boldsymbol{x})\right)^r\right)^{\frac{1}{r}}\right\} \tag{7}$$

Dubois and Prade Intersection (DPI) and Dubois and Prade Union (DPU):

$$d_j(\boldsymbol{x}) = \frac{\mu_j(\boldsymbol{x})\, p_j(\boldsymbol{x})}{\max\left\{\mu_j(\boldsymbol{x}), p_j(\boldsymbol{x}), \alpha\right\}} \tag{8}$$

$$d_j(\boldsymbol{x}) = \frac{\mu_j(\boldsymbol{x}) + p_j(\boldsymbol{x}) - \mu_j(\boldsymbol{x})\, p_j(\boldsymbol{x}) - \min\left\{\mu_j(\boldsymbol{x}), p_j(\boldsymbol{x}), (1-\alpha)\right\}}{\max\left\{\left(1-\mu_j(\boldsymbol{x})\right), \left(1-p_j(\boldsymbol{x})\right), \alpha\right\}} \tag{9}$$

Werners Intersection (WI) and Werners Union (WU):

$$d_j(\boldsymbol{x}) = \gamma \min\left\{\mu_j(\boldsymbol{x}), p_j(\boldsymbol{x})\right\} + \frac{(1-\gamma)\left(\mu_j(\boldsymbol{x}) + p_j(\boldsymbol{x})\right)}{2}. \tag{10}$$

$$d_j(\boldsymbol{x}) = \gamma \max\left\{\mu_j(\boldsymbol{x}), p_j(\boldsymbol{x})\right\} + \frac{(1-\gamma)\left(\mu_j(\boldsymbol{x}) + p_j(\boldsymbol{x})\right)}{2} \tag{11}$$

Symmetric Summations M1 (M1) and Symmetric Summations M2 (M2):

$$d_j(\boldsymbol{x}) = \frac{\mu_j(\boldsymbol{x}) + p_j(\boldsymbol{x}) - \mu_j(\boldsymbol{x})\, p_j(\boldsymbol{x})}{1 + \mu_j(\boldsymbol{x}) + p_j(\boldsymbol{x}) - 2\mu_j(\boldsymbol{x})\, p_j(\boldsymbol{x})} \tag{12}$$

$$d_j(\boldsymbol{x}) = \frac{\mu_j(\boldsymbol{x})\, p_j(\boldsymbol{x})}{1 - \mu_j(\boldsymbol{x}) - p_j(\boldsymbol{x}) + 2\mu_j(\boldsymbol{x})\, p_j(\boldsymbol{x})} \tag{13}$$

Symmetric Differences N1 (N1) and Symmetric Differences N2 (N2):

$$d_j(\boldsymbol{x}) = \frac{\max\left\{\mu_j(\boldsymbol{x}), p_j(\boldsymbol{x})\right\}}{1 + \left|\mu_j(\boldsymbol{x}) - p_j(\boldsymbol{x})\right|} \tag{14}$$

$$d_j(\boldsymbol{x}) = \frac{\min\left\{\mu_j(\boldsymbol{x}), p_j(\boldsymbol{x})\right\}}{1 - \left|\mu_j(\boldsymbol{x}) - p_j(\boldsymbol{x})\right|} \tag{15}$$

The final decision for the sample $\boldsymbol{x}$ in each of these fuzzy aggregation operations is made by choosing the maximum of w_j obtained, through the following expression,

$$\boldsymbol{x} \in w_j \quad if \quad \left(d_j\right) > \left(d_k\right) \;\; \forall d_k \mid d_k \neq d_j \tag{16}$$

3 Comparative Analysis and Performance Evaluation

To assess the validity and performance of the proposed approach we describe the tests carried out according to both processes: training and classification.

We have used a set of 26 digital aerial images acquired during May in 2006 from the Abadin region located at Lugo (Spain). They are multispectral images with 512x512 pixels in size. The images are taken at different days from an area with several natural textures. The initial training patterns are extracted from 10 images of the full set. The remainder 16 images are used for testing. The images assigned to each set are randomly selected from the 26 images available.

3.1 Unsupervised Training: Estimating the Best Partition

The first goal is to determine the number of classes that will validate the initial partition [12]. Figure 2 shows the behavior of the partition coefficient, *PC* equation (2), versus the number of clusters. It should be noted that there are two partition coefficient values exceeding the threshold value of 0.8 in our experiments, considered as appropriate. These correspond to the values of *c* for *4* and *5*, with respective values of 0.86 and 0.82. As the maximum value is obtained for $c = 4$, this value is eventually chosen as the number of clusters for classifying our images. The reminder clusters do not achieve acceptable values according to criteria set threshold 0.8.

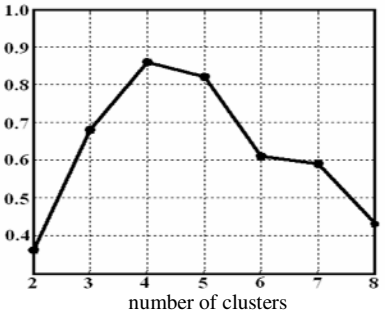


Fig. 2. Values for PC against the number of clusters

3.2 Design of a Test Strategy

The set of 26 images is split into two subsets *A* and *B* with ten and sixteen images respectively; an initial training process is carried out with *A* using FC and BP. The set *B* is classified after this initial training phase. The above mentioned parameters estimated by FC and BP are stored in *KB*. During the classification phase the new patterns are classified by FC and BP and also through the aggregations defined in equations (4) to (15) recovering the parameters stored in *KB*. To verify the performance of the base, FC, BP, and combined, equations (4) to (15), classifiers we build a ground truth for each image under the supervision of the expert human criterion considering that the number of clusters is four as obtained based on the *PC* coefficient, which agree with the expert criterion. For each class we build a binary

image, which is manually touched up until a satisfactory classification is obtained under the human supervision. Using the ground truth for each simple classifier we choose two of the best results, which are the fuzzy clustering classifier and the Bayesian classifier.

In summary, the full test process is carried out according to the following steps:

STEP 0 (initial training): for each image (from the 26 available) we perform a downsampling by 4, i.e. we obtain 26x128x128 training samples.

For $c = 2$ to $c = 8$ (maximum number of classes allowed) validate the partition by computing the partition coefficient PC through the equation (2) and determine the best partition (number of classes and centers). These classes are used for classifying the pattern samples during the next steps.

STEP 1: given the images in A, classify each pixel as belonging to a textured class, which has been identified previously, according to the FC and BC. Store the parameters in KB to be used in the next step.

STEP 2: using the set B, classify it according to FC and BC and also make a decision with the different fuzzy aggregation operations. Compute the percentage of successes according to the ground truth defined for each class at each image.

3.3 Analysis of Results

In order to verify the performance of the proposed hybrid strategy, we compare the percentage of error obtained during the classification phase for the base classifiers and the hybrid ones according to the ground truth and the four classes estimated as valid. Table 1 shows these percentages.

Table 1. Percentage of error obtained for the methods analysed

Base	FC	BP										
	17.04	17.83										
Hybrid	HU	HI	YU	YI	DPU	DPI	WU	WI	M1	M2	N1	N2
	15.42	16.68	17.02	17.10	16.13	16.52	16.6 3	16.79	16.94	16.94	31.00	19.39

From results in table 1, one can see that the error obtained with FC is comparable to that obtained with BP. Also that the best performances are obtained with the fuzzy aggregation operations, like HU, HI, DPU, DPI, WU, WI, M1 and M2. The best and worse performances are obtained by HU and N1 respectively. We can infer as a general conclusion that the hybridization improves the classification results. Exception made with N1 and N2. Figure 3 (*a*) displays an original image which is to be classified; (*b*) displays the correspondence between the classes and the color assigned to the corresponding cluster center according to a color map previously defined; (*c*) labeled image for the four clusters obtained by the hybrid HU approach. The correspondence between labels and the different textures is as follows, 1: yellow with forest vegetation, 2: blue with bare soil, 3: green with agricultural crop vegetation, 4: red with buildings and man made structures.

4 Conclusions

We propose a new hybrid classifier for natural textures. The proposed method combines two classifiers, FC and BP through fuzzy aggregation operations. The performance of the hybrid approach is compared against the base classifiers, verifying that it performs favourably in the set of aerial images tested. This approach could be applicable to other textured images even with different attributes. Also, as a future work different number of classifiers could be used by applying the associative property of fuzzy aggregations. Moreover, thanks to the proposed design, the method becomes unsupervised even though the base classifiers are supervised.

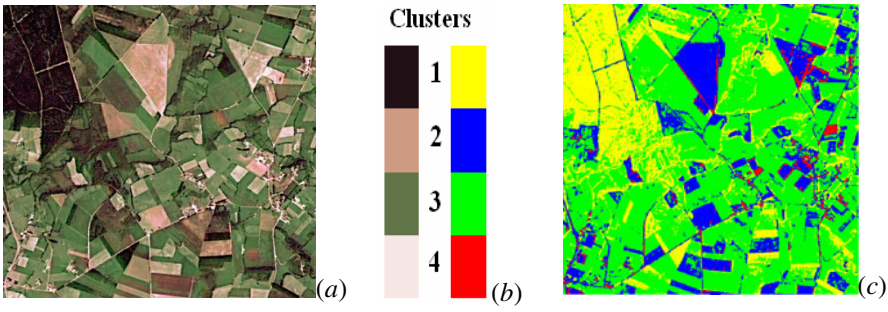


Fig. 3. (a) Original image; (b) colors and labels; (c) labeled image result of the HU

Acknowledgments. Partial funding has also been received from DPI2009-14552-C02-01 project, supported by the Ministerio de Educación y Ciencia of Spain within the Plan Nacional de I+D+i.

References

1. Drimbarean, P.F., Whelan, P.F.: Experiments in Colour Texture Analysis. Pattern Recognition Letters 22, 1161–1167 (2003)
2. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. IEEE Trans. on Pattern Analysis and Machine Intelligence 20(3), 226–239 (1998)
3. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, Chichester (2004)
4. Valdovinos, R.M., Sánchez, J.S., Barandela, R.: Dynamic and Static Weighting in Classifier Fusion. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005. LNCS, vol. 3523, pp. 59–66. Springer, Heidelberg (2005)
5. Duda, R.O., Hart, P.E., Stork, D.S.: Pattern Classification. Wiley, Chichester (2000)
6. Zimmermann, H.J.: Fuzzy Set Theory and its Applications. Kluwer Academic Publishers, Norwell (1991)
7. Guijarro, M., Pajares, G., Abreu, R.: A New Unsupervised Hybrid Classifier for Natural Textures in Images. In: HAIS, vol. 44, pp. 280–287. Springer, Heidelberg (2007)
8. 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)

9. Corchado, E., Abraham, A., Carvalho, A.C.P.L.F.D.: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)
10. Wozniak, M., Zmyslony, M.: Designing Fusers on the Basis of Discriminants - Evolutionary and Neural Methods of Training. HAIS 1, 590–597 (2010)
11. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)

A Generalization of Majority Voting Scheme for Medical Image Detectors

Henrietta Toman, Laszlo Kovacs, Agnes Jonas,
Lajos Hajdu, and Andras Hajdu

University of Debrecen
Egyetem ter 1, 4032 Debrecen, Hungary
{toman.henrietta,kovacs.laszlo.ipgd}@inf.unideb.hu,
jonasagn@gmail.com,hajdul@math.klte.hu,
hajdu.andras@inf.unideb.hu

Abstract. In this paper we propose a method for locating the optic disc (OD) in retinal images automatically using a generalization of majority voting scheme. Applying more different optic disc detectors for voting we can achieve better performance for the automatic detection system than for each individual algorithm. The location with maximum number of OD center candidates falling within a radius predefined clinically can be used to localize the OD center. In contrast to the classical voting system we can make good decision if the number of algorithms detecting the optic disc correctly is less than the half of the overall number of algorithms.

Keywords: biomedical imaging, diabetic retinopathy, majority voting.

1 Introduction

Progressive eye diseases can be caused by diabetic retinopathy (DR) which can lead to blindness, as well. One of the first essential steps in automatic grading of the retinal images is to determine the exact location of the main anatomical features, such as the optic disc and the macula. The optic disc can be considered as a bright region with circular shape. The center of the macula is called fovea which is responsible for the sharpest vision. The locations of these features play important role in making diagnosis in the clinical protocol. For the OD detection we wondered, whether organizing more individual OD detector algorithms (for example [1]) into a voting system may raise detection accuracy.

In our approach, all of the OD algorithms return with the OD center as a single pixel. We have combined the output for the center of the optic disc of each detector and considered the minimal bounding circles for all subgroups of the candidates. The radius of the circle must be less than or equal to the radius of the optic disc which is a clinically predetermined constant. The circle with maximal number of candidates is chosen for the optic disc.

From our empirical results we have found that if we choose different methods with complementary strengths for combining the detectors the overall performance of the system can substantially improve [2].

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 189–196, 2011.

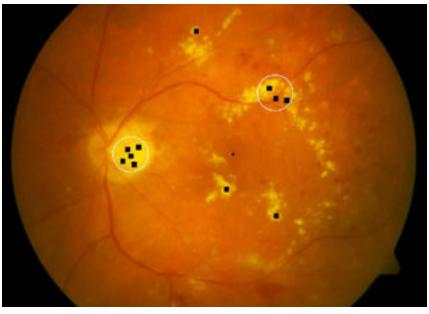


Fig. 1. Results of the different OD detection algorithms

To be able to characterize the accuracy of this combined system a corresponding theoretical model is needed. The literature offers a good support when majority voting can be applied ([3], [4]). However, in our case majority voting cannot be applied directly since besides the logical value of the votes their spatial replacement also count. In this combined system with the above mentioned voting scheme we can make a good decision even in the case when the bad candidates have majority such as in the case illustrated in Fig.1. Bad decision can be made only when a subset of bad candidates with larger cardinality than the number of good ones can be bounded by a circle with an appropriate radius. This decision suits the clinical protocol, since the OD is modelled by a disc having this radius.

This observation motivated to work out a corresponding theoretical model, where bad votes can overcome good ones only if a further condition is fulfilled which is the spatial closeness of the candidates in the above application. With this model we generalize the simple majority voting scheme, since in the case of less good votes we still have some chance to make a good decision. This generalized method can be applied to several problems corresponding to spatial location with additional constraints (e.g. detecting a certain pixel or region).

In the rest of the paper, section 2 presents the classical voting system. In section 3 we define a generalization of voting system and show some theoretical results, while section 4 discusses our results for the specific OD detection application. Section 5 gives conclusion and further recommendations.

2 Majority Voting

Let $D = (D_1, D_2, \ldots, D_n)$ be a set of classifiers, $D_i : R^k \to \Omega$ $(i = 1, \ldots, n)$ where $\Omega = (\omega_1, \omega_2, \ldots, \omega_c)$ be a set of class labels. In the majority vote method of combining classifier decisions the class label ω_i supported by the majority of the classifiers D_i is assigned to x. Most often ties are broken randomly.

As a special case, we can consider binary classifiers examined exhaustively in the literature. That is, let n be odd, $\Omega = (\omega_1, \omega_2)$ (each classifier output is a binary vector) and all classifiers have the same classification accuracy p.

An accurate class label is given by the majority vote if at least $[n/2]$ classifiers give correct answers. The majority vote method with independent classifier decisions gives an overall correct classification accuracy calculated by the binomial formula:

$$P = \sum_{k=0}^{[n/2]} \binom{n}{k} p^{n-k}(1-p)^k. \tag{1}$$

Several interesting results can be found in [5] applying the majority voting in pattern recognition. This method is guaranteed to give a higher accuracy than the individual classifiers if the classifiers are independent and $p > 0.5$.

3 The Generalization

Let $\eta = (\eta_1, \ldots, \eta_n)$ be an n-dimensional random variable. Assume that the coordinates η_i of η are independent random variables with

$$P(\eta_i = 1) = p, \quad P(\eta_i = 0) = 1 - p \quad (i = 1, \ldots, n) \ , \tag{2}$$

where $p \in [0, 1]$. Execute the experiment η independently t times, and write the outcomes in a table of size $n \times t$. (The j-th column of the table contains the realization of η in the j-th experiment $(j = 1, \ldots, t)$.) Define now the random variables $\chi_1, \ldots, \chi_t$ in the following way. If in the j-th column there are k ones then let

$$P(\chi_j = 1) = p_{nk}, \quad P(\chi_j = 0) = 1 - p_{nk} \quad (j = 1, \ldots, t) \ , \tag{3}$$

where the p_{nk}-s $(k = 0, 1, \ldots, n)$ are given numbers with

$$0 \le p_{n0} \le \cdots \le p_{nn} \le 1 \ . \tag{4}$$

These relations are also illustrated in Fig. 2.

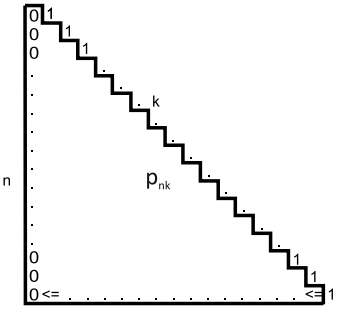


Fig. 2. The triangular matrix of the p_{nk} entries

Observe that the χ_j-s are independent. Finally, put

$$\xi = |\{j \ : \ \chi_j = 1\}| \ , \tag{5}$$

that is, ξ is the number of "good" decisions. Observe that all the individual decisions η_i $(i = 1, \ldots, n)$ are of binomial distribution with parameters (t, p). As we shall see, ξ is also of binomial distribution, with the appropriate parameters. To show this, first we need the following lemma.

Lemma 1. *For any $j = 1, \ldots, t$ we have*

$$P(\chi_j = 1) = \sum_{k=0}^{n} p_{nk} \binom{n}{k} p^k (1-p)^{n-k}. \tag{6}$$

Proof. The statement is trivial in view of the definitions of the objects involved.

We introduce the following notation: put

$$q = \sum_{k=0}^{n} p_{nk} \binom{n}{k} p^k (1-p)^{n-k}. \tag{7}$$

Lemma 2. *The random variable ξ is of binomial distribution with parameters (t, q).*

Proof. Let $k \in \{0, 1, \ldots, t\}$. Then, since the χ_j-s are independent, we have

$$P(\xi = k) = \binom{n}{k} q^k (1-q)^{n-k}, \tag{8}$$

and the statement follows.

In order to have the majority voting be "better" than the individual decisions, we need only to guarantee that $q \geq p$. The next statement yields a guideline along this way.

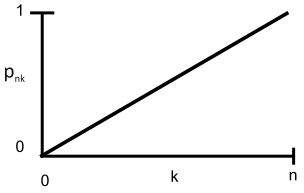
Proposition 1. *Let $p_{nk} = k/n$ $(k = 0, 1, \ldots, n)$. Then we have $q = p$, and consequently $E\xi = tp$.*

Proof. By Lemma 2 we have $E\xi = tq$. Thus we need only to show that $q = p$ whenever $p_{nk} = k/n$ $(k = 0, 1, \ldots, n)$. Indeed, using that a random variable of binomial distribution with parameters (n, p) has expected value np, in this case we have

$$q = \sum_{k=0}^{n} p_{nk} \binom{n}{k} p^k (1-p)^{n-k} = \sum_{k=0}^{n} k/n \binom{n}{k} p^k (1-p)^{n-k} = np/n = p \ . \tag{9}$$

Hence the statement follows.

As a trivial consequence we obtain the following statement.



(a) The curve of p_{nk}

	L = 3	L = 5	L = 7	L = 9
p = 0.6	0.6	0.6	0.6	0.6
p = 0.7	0.7	0.7	0.7	0.7
p = 0.8	0.8	0.8	0.8	0.8
p = 0.9	0.9	0.9	0.9	0.9

(b) System accuracy

Fig. 3. The results of the linear case

Corollary 1. *Suppose that for all* $k = 0, 1, \ldots, n$ *we have* $p_{nk} \geq k/n$. *Then* $q \geq p$, *and consequently* $E\xi \geq tp$.

We indicate the overall performance (P) of the voting system in Fig. 3 for different number of classifiers/algorithms (L) with different accuracies (p) in the linear case when $p_{nk} = k/n$.

Theorem 1. *Suppose that* $p \geq 1/2$ *and for any* $n \geq k_1 > k_2 \geq 0$ *with* $k_1 + k_2 = n$ *we have* $p_{nk_1} + p_{nk_2} \geq 1 (= (k_1 + k_2)/n)$ *and* $p_{nk_1} \geq k_1/n$; *further, that* $p_{n\frac{n}{2}} \geq 1/2$ *if* n *is even. Then* $q \geq p$, *and consequently* $E\xi \geq tp$.

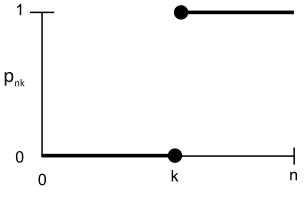
Proof. One can easily check that for any k_1, k_2 as in the statement we have

$$\begin{aligned} p_{nk_1}\binom{n}{k_1}p^{k_1}(1-p)^{n-k_1} + p_{nk_2}\binom{n}{k_2}p^{k_2}(1-p)^{n-k_2} \geq \\ \geq \frac{k_1}{n}\binom{n}{k_1}p^{k_1}(1-p)^{n-k_1} + \frac{k_2}{n}\binom{n}{k_2}p^{k_2}(1-p)^{n-k_2}. \end{aligned} \tag{10}$$

This by Proposition 1 clearly implies the statement.

As a trivial consequence we obtain the following result of Kuncheva et al [4].

Corollary 2. *Suppose that* n *is odd,* $p \geq 1/2$ *and for all* $k = 0, 1, \ldots, n$ *we have* $p_{nk} = 1$, *if* $k > n/2$, *and* $p_{nk} = 0$, *otherwise. Then* $q \geq p$, *and consequently* $E\xi \geq tp$.



(a) The curve of p_{nk}

	L = 3	L = 5	L = 7	L = 9
p = 0.6	0.6480	0.6826	0.7102	0.7334
p = 0.7	0.7840	0.8369	0.8740	0.9012
p = 0.8	0.8960	0.9421	0.9667	0.9804
p = 0.9	0.9720	0.9914	0.9973	0.9991

(b) System accuracy

Fig. 4. The results of the classical majority voting scheme

We indicate the overall performance (P) of the voting system in Fig. 4 for different number of classifiers/algorithms (L) with different accuracies (p) in the classical case.

Of particular interest is the value $P(\xi = t)$, since this expresses the probability that we make only "good" decisions. In case of an individual decision, the corresponding probability is p^t. So we need to choose the probabilities p_{nk} so that $P(\xi = t) \geq p^t$. In fact we can characterize a much more general case. For this purpose we need the following lemma, due to Gilat [6].

Lemma 3. *For any integers t and l with $t \geq 1$ and $1 \leq l \leq t$ the function*

$$f(x) = \sum_{k=l}^{t} \binom{t}{k} x^k (1-x)^{t-k} \tag{11}$$

is strictly monotone increasing on $[0, 1]$.

Note that obviously, for any $x \in [0, 1]$ we have

$$\sum_{k=0}^{t} \binom{t}{k} x^k (1-x)^{t-k} = 1 \ . \tag{12}$$

As a simple consequence of Lemma 3 we obtain the following result. Recall that the η_i-s $(i = 1, \ldots, n)$ are just "individual" random variables, of binomial distribution with parameters (t, p).

Theorem 2. *Let t and l be integers with $t \geq 1$ and $1 \leq l \leq t$. Then $P(\xi \geq l) \geq P(\eta_1 \geq l)$ if and only if $q \geq p$, i.e. $E\xi \geq tp$.*

Proof. Let t and l be as given in the statement. Then we have

$$P(\xi \geq l) = \sum_{k=l}^{t} \binom{t}{k} q^k (1-q)^{t-k} \tag{13}$$

and

$$P(\eta_1 \geq l) = \sum_{k=l}^{t} \binom{t}{k} p^k (1-p)^{t-k}. \tag{14}$$

Thus by Lemma 3 we obtain that

$$P(\xi \geq l) \geq P(\eta_1 \geq l) \tag{15}$$

if and only if $q \geq p$, and the theorem follows.

Theorem 3. *Let $\eta = (\eta_1, \ldots, \eta_n)$ be an n-dimensional random variable. We consider the joint distribution $q_{a_1,\ldots,a_n} = P(\eta_1 = a_1, \ldots, \eta_n = a_n)$. Let $p_{nk} = k/n$ $(k = 0, 1, \ldots, n)$. Then we have $E\xi = p$.*

Proof. It follows from rearranging the sums in the following way:

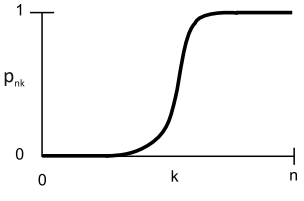
$$E\xi = \sum_{k=0}^{n} \sum_{a_1+\ldots+a_n=k} k/n \cdot q_{a_1,\ldots,a_n} = \\ 1/n \sum_{i=1}^{n} \sum_{a_i=1} q_{a_1,\ldots,a_n} = 1/n \sum_{i=1}^{n} P(\eta_i = 1) = p. \tag{16}$$

4 The Specific Application

We have already considered two different types of p_{nk} matrices which satisfy the expected properties. The system accuracy was characterized in both cases.

The first case was of the linear p_{nk}, while the second matrix was determined by the classical majority voting scheme.

The last example of the matrix p_{nk} is motivated by our medical imaging application. In this case the behavior of p_{nk} in variable k for a given n and the system accuracy are illustrated in Fig 5.



(a) The curve of p_{nk}

	L = 3	L = 5	L = 7	L = 9
p = 0.6	0.8208	0.8390	0.8895	0.9247
p = 0.7	0.9163	0.9373	0.9658	0.9823
p = 0.8	0.9728	0.9850	0.9942	0.9980
p = 0.9	0.9963	0.9988	0.9997	0.9999

(b) System accuracy

Fig. 5. The results of our application

Fig. 5 shows that the p_{nk} increase exponentially in k for a given n. This follows from the results of [7] saying the probability that the diameter of a point set is not less than a given constant is decrease exponentially if the number of points tends to infinity. Note that, this diameter corresponds again to the radius of the OD defined by the clinical protocol.

5 Conclusion

We worked out a new theoretical model that enables the investigation of majority voting systems being more general than the simple majority voting scheme. In our specific application better overall system accuracy is achieved than in case of individual algorithm. Same results are expected for all image processing problems where the algorithms vote with a single pixel or range as output. In our application adding a new algorithm to the system seems to be very effective because of the exponential behavior of the distribution function of the diameter. The full characterization of the participating algorithms to achieve the best

system performance is still an open issue. The essential criterion for the selection of the algorithms to be combined is that $p > 1/2$ for its accuracy.

A natural factor as for the accuracy of the system is the dependence of the algorithms. Though this paper concentrates on the independent case it can be shown that the accuracy can drop/raise based on the dependencies of the algorithms similarly to the majority voting case ([8]). To tune also our system it will be a future research to see how the accuracy can be raised (to approach the pattern of success) by removing/adding algorithms from/to the existing system in consideration to individual accuracies and dependencies. The pattern of success and failure is a useful information in clinical systems since they characterize the expected value of the system error and the boundary of the system accuracy.

Acknowledgments. This work was supported in part by the Janos Bolyai grant of the Hungarian Academy of Sciences, and by the TECH08- 2 project DRSCREEN- Developing a computer based image processing system for diabetic retinopathy screening of the National Office for Research and Technology of Hungary (contract no.: OM-00194/2008, OM-00195/2008, OM-00196/ 2008). Research is supported in part by the OTKA grants (K67580, K75566) and by the TÁMOP 4.2.1./B-09/1/KONV-2010-0007 project. The project is implemented through the New Hungary Development Plan, cofinanced by the European Social Fund and the European Regional Development Fund.

References

1. Mahfouz, A.E., Fahmy, A.S.: Ultrafast Localization of the Optic Disc Using Dimensionality Reduction of the Search Space. Med. Image Comput. Assist. Interv. 12, 985–992 (2009)
2. Harangi, B., Qureshi, R.J., Csutak, A., Peto, T., Hajdu, A.: Automatic Detection of the Optic Disc Using Majority Voting in a Collection of Optic Disc Detectors. In: 7th IEEE International Symposium on Biomedical Imaging, pp. 1329–1332. IEEE Press, Rotterdam (2010)
3. Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 993–1001 (1990)
4. Kuncheva, L.I.: Combining Pattern Classifiers, Methods and Algorithms. John Wiley & Sons, Inc., New Jersey (2004)
5. Lam, L., Suen, C.Y.: Application of Majority Voting to Pattern Recognition: An Analysis of Its Behavior and Performance. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans 27, 553–568 (1997)
6. Gilat, D.: Monotonicity of a Power Function: An Elementary Probabilistic Proof. The American Statistician 31, 91–93 (1977)
7. Appel, M.J., Najim, C.A., Russo, R.P.: Limit Laws for the Diameter of a Random Point Set. Adv. Appl. Probab. 34(1), 1–10 (2002)
8. Altincay, H.: On Naive Bayesian Fusion of Dependent Classifiers. Pattern Recognition Letters 26, 2463–2473 (2005)

An Efficient Hybrid Classification Algorithm – An Example from Palliative Care

Tor Gunnar Houeland and Agnar Aamodt

Department of Computer and Information Science,
Norwegian University of Science and Technology,
NO-7491 Trondheim, Norway
{houeland,agnar}@idi.ntnu.no

Abstract. In this paper we present an efficient hybrid classification algorithm based on combining case-based reasoning and random decision trees, which is based on a general approach for combining lazy and eager learning methods. We use this hybrid classification algorithm to predict the pain classification for palliative care patients, and compare the resulting classification accuracy to other similar algorithms. The hybrid algorithm consistently produces a lower average error than the base algorithms it combines, but at a higher computational cost.

Keywords: hybrid reasoning systems, classifier combination, case-based reasoning, random decision trees.

1 Introduction

Case-based reasoning (CBR), including instance-based methods, represents a unique approach to learning and problem solving compared to generalization-based methods. It is therefore often a choice of one method in a hybrid system, complementary to generalization-based and inductive methods. Examples include using an ensemble of different inductive methods to perform adaptation in CBR [12], and a neural network approach for adaptation, revision, and retainment of cases [5]. As a lazy learning method that postpones the generalization step until problem solving time [1], CBR has the advantage of including contextual information that an eager approach would not have access to, thereby adapting the reasoning to the particular characteristics of the problem to solve. Eager methods, on the other hand, have the advantage that parts of the problem solving behaviour can be precomputed during training, which enables reduced storage space and faster query processing.

A path of our research is to explore the combination of model-based reasoning, starting from a predefined model that make some top-down commitments, with case-based reasoning that make very few high level commitments but rather grows its knowledge base (case base) in a bottom-up fashion. An example is the combination of Bayesian Networks with case-based reasoning [4]. In this paper we examine a hybrid approach that uses a modified version of an eager method,

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 197–204, 2011.

Random Decision Trees (RDT), that can be partially precomputed and partially adapted to the particular problem query.

As of today, there is no consensus about the set of classes that should be used for pain classification in palliative care [9]. Our domain is open and changing, which is why we study methods of machine learning and decision support that are able to produce useful results without making very strong commitments about the domain.

In an earlier study we examined the problem of determining case similarity in our palliative care domain, and created a hybrid RDT approach to locate the most similar case in the case base [7]. In the work presented here we extend our approach by developing algorithms for classifying cases. We do this by predicting the *average pain* and *worst pain* values for the third week after the first consultation, based on the information collected for the first two weeks. These values are important because the objective is to minimize the patient's pain, and the doctor's approach for relieving the patient's pain is applied in full during the second week. In the third and following weeks, the patient is mainly observed and pain medication is modified according to needs.

In the next section we review some earlier relevant research, which is followed in section 3 by a description of our RDT-based experiment and the algorithms used in the experiment. In section 4 we compare the algorithms and their parameters and discuss empirical results from running the algorithms on a case base of palliative care patients. Concluding remarks end the paper.

2 Related Research

Studies of ensembles of random decision trees have been extensive. Among the most well-known is the Random Forest (RF) classifier [3] which grows a number of trees based on bootstrap samples of the training data. For each node of a tree, m variables are randomly chosen and the best split based on these m variables is calculated based on the bootstrap sample. Each decision tree results in a classification and is said to cast a vote for that classification. The ensemble classifier returns the class that receives the most votes. RF can also compute *proximities* between pairs of cases that can be used for clustering and data visualization, and hence as similarity measures for case-based reasoning.

In a thorough study of ensemble method types it was found that the performance of an ensemble learning approach varies substantially across applications. Bian et. al. [2] studied homogeneous and heterogeneous ensembles and found connections between diversity and performance, and an increased diversity for heterogeneous ensembles.

A contribution to the analysis of the laziness vs. eagerness distinction, which corresponds with the distinction between global and local approximations to the target function, was made by Hendrickx and den Bosch [6]. They studied several hybrid methods as well as their single components. The analysis showed that the k-NN method outperformed the eager methods, while the best hybrid methods outperformed any single methods on combined generalization performance and

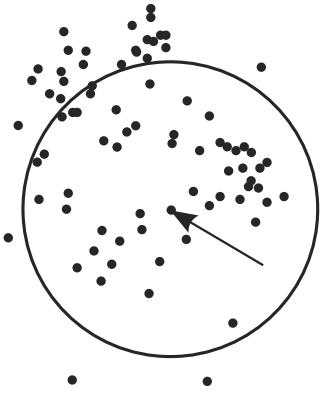


Fig. 1. Using a similarity-based local case subset as training data. A set of neighbors of the marked problem query to solve are shown as the cases that lie within the circle. That set is used to train an independent learning algorithm.

statistical error bias. A combined approach for optimizing the combined learning and classification time of lazy and eager learners was developed by Mohebpour et.al. [11], a problem also addressed by Veloso and Meira [14].

A particular problem relates to the utility of learned knowledge. The "utility problem" occurs when additional knowledge learned decreases a reasoning system's performance instead of increasing it [10,13]. Theoretically this will always occur in a CBR system when the system's case base increases without bound. The utility problem is not necessarily observed in practice for real-world CBR systems with moderately-sized case bases, however.

Based on one of our own studies [8], we suggest that the usefulness of an optimization should be measured by the effect it has on the reasoning system's overall utility. We measured an example system's total solution time to show that case base size reduction methods can be counterproductive because the methods were more computationally demanding than simply reasoning using the larger unreduced case base.

3 Random Decision Tree Classification Experiment

The hybrid random decision tree (RDT) algorithm presented here is an approach to combining machine learning methods with case-based reasoning. We retrieve the most similar half of the available cases using a domain-specific relevance measure (a general illustration of this approach is shown in figure 1). We then run our RDT algorithm as a computationally efficient machine learner using this subset of cases as training data. This approach combines the lazy and locally specific characteristics of a CBR retrieval with the more eager and global characteristics often seen in traditional machine learning algorithms.

In the presented research we expand the use of our RDT algorithm from being a pure similarity measure to also predicting the classification of unseen cases.

As part of its internal computations, each decision tree in our algorithm is partitioning the cases in the case base between its leaf nodes. This is conceptually similar to how indexing trees used for efficient retrieval in CBR are constructed. We exploit this insight to create a classification algorithm where each tree classifies a new problem query based on the previous cases that lead to the same leaf node as the new problem query.

If each tree classifies cases as the arithmetic mean of the classification of previous cases in the same leaf node, and the average of each tree is used as the combined classification, then it is not necessary to enumerate the specific subsets of cases present in each leaf node. It is sufficient to know *how many times* each case shares a leaf node with the problem query, and then the combined classification can be determined by taking the weighted average, where each case is assigned a weight equal to the number of times it shares a leaf node with the problem query.

The number of times a case shares a leaf node with the problem query is precisely the *proximity* of the case, for which we have previously developed an efficient computational method while exploring the use of RDTs to determine similarity [7].

Using this proximity-weighted averaging approach, we have implemented a purely RDT-based classifier and a hybrid RDT+CBR classifier. We explore their characteristics related to the palliative pain classification domain. For comparison purposes we also test a k-NN classifier corresponding to the CBR part of the hybrid, and a simple and very fast algorithm based only on averaging. We compare the results obtained from these algorithms according to their computational complexity.

Our data set consists of 1486 cases with numerical features based on patients in the palliative care domain. The problem description we use for input queries to be solved consists of 55 numerical features based on measurements and classifications obtained during the first two weeks after the first consultation. Examples of these features include the patient's age, the reported average pain for week 1 on a scale from 0-10, the total opioid dose given as pain relief for week 2 as a floating point number, and similar features for other aspects such as insomnia, cognitive functioning and use of antidepressants. As the solution to predict we use 2 classifications related to the patient's pain for the third week: the reported *average pain* and *worst pain* on scales from 0-10.

3.1 Algorithms

Computed-Average computes the mean *average pain* and *worst pain* values based on the cases encountered so far, and uses these computed means as the predicted classifications for the new problem query. It is a simple and fast algorithm which only learns from the problem solutions. It performs this limited task well, and is used as a baseline comparison for the other algorithms which attempt to also learn domain knowledge from the more complicated problem descriptions.

CBR-k-NN selects the average of the k most similar previously encountered cases. Similarity is measured using a simple CBR-Difference-Measure function that was provided as a rough relevance estimate. This estimate is based on differences in 8 values in the data set that correspond to the variables a domain expert considers most important. For $k = 1$ this is the same as retrieving and copying the solution from the most similar case, while $k \geq 2$ performs averaging as a simple and knowledge-lean multi-case adaptation step during reuse.

N-RandomTrees-Classifier is based on our presented approach for classification by efficiently evaluating random decision trees on case subsets. N trees are grown and the *average pain* and *worst pain* values are predicted as the average of all cases in the case base weighted by their computed proximity to the problem query.

N-Hybrid-Classifier is our hybrid combination of the CBR relevance measure and using our RDT algorithm for classification. For every input problem query, the CBR-Difference-Measure function is used to narrow the case base down to the most similar half. Then N trees are used to compute the average and worst pain as in the N-RandomTrees-Classifier algorithm, but based only on the cases from this most relevant half of the case base.

4 Results and Discussion

To achieve a fair comparison we generate 10 versions of the input where the same patient cases are used, but in 10 different randomly shuffled orders. We evaluate the algorithms by their average result from each of these modified case bases. We use this approach because the results of a single run-through of the case base can vary, due to intrinsic randomness in the RDT-based algorithms and differences caused by the order in which the cases are presented. For each algorithm we measure the root mean square error (RMSE) for solving each of the 10 permuted case bases, and report the average RMSE value.

Figure 2 shows the measured average root mean square error for the different algorithms, compared according to the time (computational resources) required. The result for the Computed-Average algorithm is shown as a single point, as there is no varying parameter and the execution time for a given set of inputs remains constant apart from small random fluctuations in the computing environment. The exact time required depends on the type of computing device that is used to run the algorithms, but we focus on the relative differences between these algorithms which is primarily determined by their computational complexity.

The results for CBR-k-NN are not as sensitive to the exact value of k as an initial reading of the graph might suggest, because the value of k has a relatively small effect on the time required to run the algorithm. In fact the visible line for CBR-k-NN in the graph spans from around $k = 5$ to $k = 1000$.

Additional details are shown in table 1, with numerical values for a subset of the results. The results shown in the table are marked as points in figure 2.

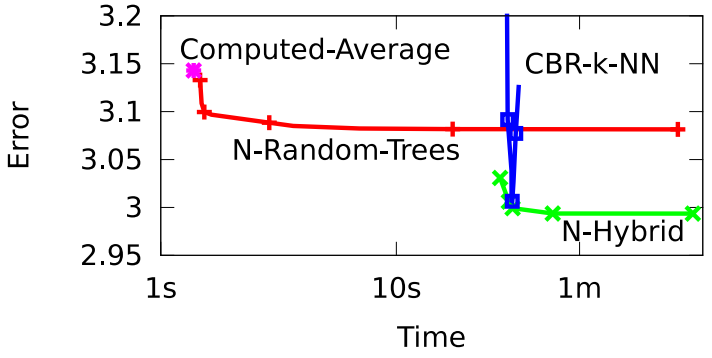


Fig. 2. Experimental results for the different algorithms and parameters, compared to the computational resources required. (Lower error and faster time is better.)

The underlying CBR-Difference-Measure function is not in itself particularly potent as a direct similarity measure. CBR-1-NN produces an error of 4.14. This is worse than a trivial classifier that always predicts 5 as the solution, which produces an error of 3.62 using the same experimental setup. However, the variables identified by the domain expert are indeed relevant, as a completely random similarity measure that retrieves a case at random produces an error of 4.46.

This indicates that the similarity measure is helpful for locating the most relevant cases, but that predicting the pain values based on only a single similar patient is unlikely to work well in this domain. A relatively large k value of around 75 produces the best result for the CBR-k-NN algorithm in this experiment.

For our RDT approaches a higher number of trees N produces better results. Unlike how k affects CBR-k-NN, there is no particular sweet spot for N in either the RDT trees or the hybrid approach above which the results start deteriorating. However, the improvements flatten out to become negligible compared to the increase in computational resources required when using more than around 1000 trees.

N-Hybrid-Classifier has the lowest overall error but a comparatively high computation costal, while Computed-Average and N-Random-Trees-Classifier are good choices to produce results very quickly.

This illustrates an important trade-off between speed and accuracy when choosing a classifier. In this experiment, our approach to combining lazy and eager classifiers to make a hybrid classifier produced better predictions, but at an increased computational cost. Whether the increased accuracy is worth the additional complexity and increased resource cost depends on the exact application and usage of the reasoning system. Given a time limit for a particular application, the algorithm that produces the best results can e.g. be determined as the lowest line at that point in a graph such as the one shown in figure 2.

Table 1. Numerical results for our algorithms in the palliative care domain, showing the computation time required and the average root mean square error

Algorithm	**Time**	**Error**
COMPUTED-AVERAGE	1.4 seconds	3.14
CBR-1-NN	26 seconds	4.14
CBR-10-NN	30 seconds	3.09
CBR-75-NN	31 seconds	3.01
CBR-500-NN	32 seconds	3.08
1-RANDOM-TREES-CLASSIFIER	1.5 seconds	3.13
10-RANDOM-TREES-CLASSIFIER	1.6 seconds	3.10
100-RANDOM-TREES-CLASSIFIER	2.9 seconds	3.09
1000-RANDOM-TREES-CLASSIFIER	17 seconds	3.08
10000-RANDOM-TREES-CLASSIFIER	150 seconds	3.08
1-HYBRID-CLASSIFIER	28 seconds	3.03
10-HYBRID-CLASSIFIER	30 seconds	3.01
100-HYBRID-CLASSIFIER	31 seconds	3.00
1000-HYBRID-CLASSIFIER	46 seconds	2.99
10000-HYBRID-CLASSIFIER	180 seconds	2.99

5 Conclusions and Further Research

In this paper we have presented an approach for classifying unseen cases in the palliative care domain by extending our efficiently computable random decision tree (RDT) algorithm. We have developed methods for predicting the *average pain* and *worst pain* values for palliative care patients. We used a case-based k-NN method using a domain-specific relevance measure, a knowledge-lean implementation of our RDT method and a hybrid combination of the relevance measure and the RDT approach. The base RDT approach produced results very quickly, while the hybrid approach produced better results than either of the base algorithms at a comparable computational cost to running the k-NN method.

In the palliative care domain, where patients receive treatment over several months and a better result can potentially result in reduced suffering, using the best possible algorithm is usually worthwhile. However, in this domain, increasing the parameter for the number of trees in the hybrid algorithm above around 1000 increases the computational cost with negligible improvements in accuracy.

In our ongoing and future work, we are experimenting with using meta-level reasoning as part of the problem solving process. Our goal is to automatically determine which algorithm produces the best results for a given data set, and to use that algorithm for solving future problem queries.

Acknowledgments. This research is partly conducted within the project TL-CPC (Transactional Research in Lung Cancer and Palliative Care), a nationally funded project in cooperation with the Medical Faculty of our university and the St. Olav Hospital in Trondheim.

We wish to thank Cinzia Brunelli for providing the data set, Anne Kari Knudsen for interpreting and analysing the data from a clinical perspective, and Tore Bruland for his analysis of the data from a data structure and machine learning perspective.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 7(1), 39–59 (1994), http://portal.acm.org/citation.cfm?id=196108.196115
2. Bian, S., Wang, W.: On diversity and accuracy of homogeneous and heterogeneous ensembles. Int. J. Hybrid Intell. Syst. 4, 103–128 (2007), http://portal.acm.org/citation.cfm?id=1367006.1367010
3. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001), http://dx.doi.org/10.1023/A:1010933404324
4. Bruland, T., Aamodt, A., Langseth, H.: Architectures integrating case-based reasoning and bayesian networks for clinical decision support. In: Shi, Z., Vadera, S., Aamodt, A., Leake, D. (eds.) IIP 2010. IFIP Advances in Information and Communication Technology, vol. 340, pp. 82–91. Springer, Heidelberg (2010)
5. Corchado, J.M., Lees, B., Aiken, J.: Hybrid instance-based system for predicting ocean temperatures. International Journal of Computational Intelligence and Applications, 35–52 (2001)
6. Hendrickx, I., van den Bosch, A.: Hybrid algorithms with instance-based classification. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 158–169. Springer, Heidelberg (2005)
7. Houeland, T.G.: An efficient random decision tree algorithm for case-based reasoning systems. In: FLAIRS Conference. AAAI Press, Menlo Park (to appear 2011)
8. Houeland, T.G., Aamodt, A.: The utility problem for lazy learners - towards a non-eager approach. In: Bichindaritz, I., Montani, S. (eds.) ICCBR 2010. LNCS, vol. 6176, pp. 141–155. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-14274-1, doi:10.1007/978-3-642-14274-1
9. Knudsen, A., Aass, N., Fainsinger, R., Caraceni, A., Klepstad, P., Jordhy, M., Hjermstad, M., Kaasa, S.: Classification of pain in cancer patients a systematic literature review. Palliative Medicine 23(4), 295–308 (2009), http://pmj.sagepub.com/content/23/4/295.abstract
10. Minton, S.: Quantitative results concerning the utility of explanation-based learning. Artif. Intell. 42(2-3), 363–391 (1990)
11. Mohebpour, M.R., Adznan, B.J., Saripan, M.I.: Grid Base Classifier in Comparison to Nonparametric Methods in Multiclass Classification. Pertanika J. Sci. & Technol. 18(1), 139–154 (2010)
12. Policastro, C., Delbem, A., Mattoso, L., Minatti, E., Ferreira, E., Borato, C., Zanus, M.: A Hybrid Case Based Reasoning Approach for Wine Classification. In: ISDA, pp. 395–400 (2007)
13. Smyth, B., Cunningham, P.: The utility problem analysed - a case-based reasoning perspective. In: Proceedings of the Third European Workshop on Case-Based Reasoning, pp. 392–399. Springer, Heidelberg (1996)
14. Veloso, A., Meira, Jr., W.: Eager, lazy and hybrid algorithms for multi-criteria associative classification. In: Proceedings of the Data Mining Algorithms Workshop, Uberlandia, MG (2005)

An Effective Feature Selection Algorithm Based on the Class Similarity Used with a SVM-RDA Classifier to Protein Fold Recognition

Wiesław Chmielnicki[1] and Katarzyna Stąpor[2]

[1] Jagiellonian University, Faculty of Physics, Astronomy and Applied Computer Science,
[2] Silesian University of Technology, Institute of Computer Science
wieslaw.chmielnicki@uj.edu.pl
http://www.fais.uj.edu.pl

Abstract. Feature selection is very important procedure in many pattern recognition problems. It is effective in reducing dimensionality, removing irrelevant data, and increasing accuracy of a classifier. In our previous work we propose a classifier combining the support vector machine (SVM) classifier with regularized discriminant analysis (RDA) classifier used to protein fold recognition problem. However high dimensionality of the feature vectors and small number of samples in the training data set caused that the problem is ill-posed for an RDA classifier and the feature selection is crucible for the accuracy of the classifier. In this paper we propose a simple and effective algorithm based on the class similarity which solves our problem and helps us to achieve very good acuracy on a real-world data set.

Keywords: Feature Selection, Support Vectore Machine, Statistical classifiers, RDA classifier, protein fold recognition.

1 Introduction

Feature selection is defined as the process of selecting most discriminatory subset of the features. The procedure aims to identify and remove as much irrelevant and redundant information as possible. This problem is NP-hard. So, the optimal solution usually cannot be found unless exhaustive search in the feature space is possible [5]. The dimensionality of the feature vectors usually cause that such approach is completely inefficient and often impossible.

In our previous paper [4] we presented a classifier which combined the support vector machine (SVM) classifier with statistical regularized discriminant analysis (RDA) classifier. The hybrid classifier was used to the protein fold recognition problem. However high dimensionality of the feature vectors (126D) and small number of samples in the training data set (385 samples in the set) caused that the problem is ill-posed for an RDA classifier. In [4] we propose an exhaustive search feature selection algorithm. It will be not possible to check all combinations of 126 features, but we profit from the fact that the features consist of six subsets (21 values each). So, we check all possible combinations of these subsets.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 205–212, 2011.

This approach was sufficient to solve our problem, but it was not very effective. In this paper we present a feature selection algorithm based on the class similarity which effectively helps us to achieve good accuracy of the RDA classifier. The accuracy of the combined SVM-RDA classifier is enhanced as well.

Protein structure (fold) prediction is one of the most important goals pursued by bioinformatics. There are several machine-learning methods to predict the protein folds from amino acids sequences proposed in literature. Ding and Dubchak [6] experimented with support vector machine (SVM) and neural network (NN) classifiers. Shen and Chou [20] proposed ensemble model based on nearest neighbour. A modified nearest neighbour algorithm called K-local hyperplane (HKNN) was used by Okun [18]. Nanni [17] proposed ensemble of classifiers: Fishers linear classifier and HKNN classifier.

The rest of this paper is organized as follows: Section 2 introduces the database and the feature vectors, Section 3 presents the hybrid classifier, Section 4 decribes proposed feature selection algorithm, Section 5 presents our experiments and finally in Section 6 the experimental results and conclusions are discussed.

2 The Database and Feature Vectors

Using machine-learning methods entails the necessity to find out databases with representation of known protein sequences and its folds. Then this information must be converted to the feature space representation.

In experiments described in this paper two data sets derived from SCOP (Structural Classification of Proteins) database are used. The detailed description of these sets can be found in Ding and Dubchak [6]. The training set consists of 313 protein sequences and the testing set consists of 385 protein sequences. The training set was based on PDB_select sets (Hobohm et al. [12], Hobohm and Sander [13]) where two proteins have no more than 35% of the sequence identity. The testing set was based on PDB–40D set developed by Lo Conte et al. [16] from which representatives of the same 27 largest folds are selected. The proteins that had higher than 35% identity with the proteins of the training set are removed from the testing set.

In our experiments the feature vectors developed by Ding and Dubchak [6] were used. These feature vectors are based on six parameters: Amino acids composition (C), predicted secondary structure (S), Hydrophobity (H), Normalized van der Waals volume (V), Polarity (P) and Polarizability (Z). Each parameter corresponds to 21 features except Amino acids composition (C), which corresponds to 20 features. The data sets including these feature vectors are available at `http://ranger.uta.edu/~chqding/protein/`. For more concrete details, see Dubchak et al. [7]. The feature vector was slightly changed. The length of the amino acid sequence was added to the Amino acids composition (C) vector, so now the C vector has also $20 + 1 = 21$ features. Therefore the full feature vector (C, S, H, V, P, Z) counts 6 x 21 = 126 features.

3 The SVM-RDA Classifier

The hybrid SVM-RDA classifier combines two different approaches to the classification task: discriminative and generative. The characteristics of these kinds of classifiers differs in several respects. The detailed discussion can be found in [15]. Our goal was to benefit from the advantages of both types of the classifiers.

3.1 The SVM Classifier

The Support Vector Machine (SVM) is a well known large margin classifier proposed by Vapnik [21]. The basic concept behind the SVM classifier is to search an optimal separating hyperplane, which separates two classes. The decision function of the binary SVM is:

$$f(x) = sign\Big(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\Big) \ , \tag{1}$$

where $0 \leq \alpha_i \leq C, i = 1, 2, \ldots, N$ are nonnegative Lagrange multipliers, C is a cost parameter, that controls the trade off between allowing training errors and forcing rigid margins, x_i are the support vectors and $K(x_i, x)$ is the kernel function. In our experiments we used the radial basis function (RBF) kernel:

$$K(x_i, x) = -\gamma \|x - x_i\|^2, \gamma > 0 \ , \tag{2}$$

3.2 The RDA Classifier

Quadratic discriminant analysis (QDA) [9] models the likelihood of class as a Gaussian distribution and then uses the posterior distributions to estimate the class for a given test vector. This approach leads to the discriminant function:

$$d_k(x) = (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log |\Sigma_k| - 2 \log p(k) \ , \tag{3}$$

where x is the test vector, μ_k is the mean vector, Σ_k covariance matrix and $p(k)$ is prior probability of the class k. The values of Σ_k and μ_k are replaced in formula (3) by its estimates $\hat{\Sigma_k}$ and $\hat{\mu_k}$.

However, when the number of training samples n is small compared to the number of dimensions of the training vector the covariance estimation can be ill-posed. The approach to resolve the ill-posed estimation is to regularize covariance matrix. The covariance matrix Σ_k can be replaced by their average i.e. $\hat{\Sigma} = \sum \hat{\Sigma_k} / \sum \hat{N_k}$ which leads to Linear Discriminant Analysis (LDA). However, it is very limited approach so in regularized discriminant analysis (RDA) (Friedman [8]) each covariance matrix can be estimated as:

$$\hat{\Sigma_k}(\lambda) = (1 - \lambda)\hat{\Sigma_k} + \lambda \hat{\Sigma} \ , \tag{4}$$

where $0 \leq \lambda \leq 1$. The parameter λ controls the degree of shrinkage of the individual class covariance matrix estimate toward the pooled estimate.

3.3 The Combined SVM-RDA Classifier

The SVM is a binary classifier but the protein fold recognition is a multi-class problem. There are many methods proposed in literature to deal with this problem. In our experiments we used the well-konwn one-versus-one strategy. In this strategy the binary classifiers are trained between all possible pairs of classes. Every binary classifier votes for the preferred class and in this way the voting table is created. The class with maximum number of votes is recognized as the correct class. However some of these binary classifiers are unreliable. The votes from these classifiers influence the final classification result. In our method there is a strategy presented to assign a weight to each vote.

It is easy to notice that the bigger value of the discriminant function for the class the smaller probability that the vector belongs to this class. So we propose to define a weight for each vote:

$$w(x,i) = \left(1 - \frac{d_i(x) - d_{min}(x)}{d_{min}(x)}\right)^3, \tag{5}$$

where x is a feature vector, i is the index of the class to which classifier assigns a sample, $d_i(x)$ the value of the discriminant function as defined in (3), and $d_{min}(x) = \min\{d_k(x)\}$, $k = 1, 2, \ldots, n$. The more detailed discussion can be found in [4].

4 Feature Selection Algorithm

There are many feature selection algorithms described in literature. However we look for a simple, but effective approach which will solve our problem with the RDA classifier. We check several methods for example [11] based of mutual correlation and some methods described in [10]. We also try a method proposed in [14]. However, all these methods are not suitable or not efficient in our case.

So, our first approach was to find some kind of measure of class similarity, which can elimitate the noisy features i.e. the features which cause that the samples of the same class are not similar. The measure should be fast and easy to compute. The simplest is the average Euclidean distance between all samples of the same class i.e.:

$$Dist(k,f) = \frac{\sum_{i=1}^{N_k} \sum_{j=i+1}^{N_k} \|x_i - x_j\|^2}{N_k * (N_k - 1)/2}, \tag{6}$$

where k is the class index, N_k number of samples of the class k and x_i, x_j are the feature vectors representing the i-th and the j-th sample and f is a feature subset.

Then we calculate the average value of the $Dist(k,f)$ for all classes as defined below:

$$AvgDist(f) = \frac{\sum_{k=1}^{M} Dist(k,f)}{M}, \tag{7}$$

where f is a feature subset, M the number of classes and $Dist(k,f)$ the distance defined in (6).

```
function FeatureSelect(Features[])
{
        Result = 0;
        for (int i = 126; i > 0; i--) {
                List = GetDist(i, Features);
                SortFeatureList(List);
                for (int j = 0; j < i; j++) {
                        Features[List[j]] = true;
                        Res = CrossValidateRda(0.7, Features);
                        if (Res > Result) {
                                Result = Res;
                                break;
                        }
                        Features[List[j]] = false;
                }
                if (NoFeatureAdded()) break;
        }
}
```

Fig. 1. The pseudo-code for the CS-SFS algorithm

It is clear that if the value of $AvgDist(f)$ is smaller then the samples of the same class are closer in a sense of the Euclidean distance. So in our algorithm we propose to prefer the features subsets which have this value as small as possible. The algorithm Class Similarity Sequential Forward Selection (CS-SFS) we propose is presented in figure 1. This algorithm is much better than simple brute force method used in our previous paper [4] or the standard Sequential Forward Selection (SFS) algorithm.

However, it has certain drawback. The measure defined in (7) tells us nothing about how a feature subset distinguish between different classes. It simply defines some kind of similarity between the samples of the same class. There can be a feature subset in which all samples are very similar.

To avoid this drawback we define a measure of similarity to all other classes as presented below.

$$Dist_{others}(k, f) = \frac{\sum_{i=1}^{N_k} \sum_{j \neq i}^{N} \|x_i - x_j\|^2}{N_k * (N - N_k)}, \tag{8}$$

and similarly

$$AvgDist_{others}(f) = \frac{\sum_{k=1}^{M} Dist_{others}(k, f)}{M}, \tag{9}$$

So, the best feature subset will be the one with minimal value of $Dist(f)$ and maximal value of $Dist_{others}(f)$. It is impossible to find a feature subset that fulfills both criteria. In our algorithm Class Similarity Others Sequential Forward Selection (CSO-SFS) we sort feature subsets: increasingly using $Dist(f)$ criterion and decreasingly using $Dist_{others}(f)$ criterion. Now, we have two lists: list one in which high position means that the samples of the same class are very similar for a given feature subset f, and list two in which high position means that the samples of every class are very dissimilar to the samples from any other class for a given feature subset f. Then we create the third list (increasingly sorted) which contains the sums of the positions of the feature subsets on the list one and the list two.

```
function FeatureSelect(Features)
{
    Result = 0;
    for (int i = 126; i > 0; i--) {
        List1 = GetDist(i, Features);   List2 = GetDistOthers(i, Features);
        SortIncreasingly(List1);        SortDecreasingly(List2);
        for (int FeatureNr = 0; FeatureNr < 126; FeatureNr++)
            List3[FeatureNr]= GetPosition(FeatureNr, List1)+GetPosition(FeatureNr, List2);
        SortIncreasingly(List3);
        for (int j = 0; j < i; j++) {
                Features[List3[j]] = true;
                Res = CrossValidateRda(0.7, Features);
                ...
                Features[List3[j]] = true;
        }
    }
}
```

Fig. 2. The pseudo-code for the CSO-SFS algorithm

It is clear that the feature subsets which are high on the third list must be relatively high on the lists one and two. It means that, if we use these feature subsets, the samples of the same class are relatively similar to each other and relatively dissimilar to the samples from all other classes. The pseudo-code for this algorithm is presented on the figure 2.

5 Experiments

Before we could start with our experiments we have to choose some parameters. However, to avoid overfitting, the cross-validation procedure must be used. We use k-fold cross-validation with k = 7, because there are at least seven samples of each class in the training data set.

In our experiments we used an SVM classifier with the RBF kernel, so the parameters C and γ must be chosen. It was done using a cross-validation procedure on the training data set. We used a grid-search algorithm with values $C = 2^{-1}, 2^0, 2^1..., 2^{10}$ and $\gamma = 0.025, 0.05, 0.1, ..., 6.4$ The best recognition ratio was achieved using parameter values $\gamma = 0.1$ and $C = 128$. All values of the feature vectors are scaled to the range [-1; +1] before using an SVM classifier to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges.

The next step was to find the best parameter value for regularization for an RDA classifier. This value must be experimentally chosen using cross-validation procedure on the training data set We checked $\lambda = 0.0, 0.1, 0.2, ..., 1.0$ and find that $\lambda = 0.7$ gave the best results. Then we used the feature selection algorithms described in Section 4 and we obtained four feature sets 25D – 63D see table 1. The RDA classifier is used with these four feature sets and using formula (3) we obtained four different weight sets.

Finally the binary SVM classifiers are trained. They used 126D full feature vectors scaled to the $[-1, +1]$ interval. Next all these binary classifiers are used on the testing data set and the voting table is created. Then the weighs obtained using the RDA classifier and the feature sets are applied to the voting tables. Now the samples with maximum number of votes (which is now a real number) is classified as the correct class.

6 Results and Conclusions

In this paper we present a feature selection algorithm based on the class similarity. The algorithm was tested on a real-world database. As we can see from table 1 the algorithm works for the RDA and for the SVM-RDA classifier as well. We can compare this with the result of the SVM classifier (on full 126D feature set) which is 57.9%. The comparison with other methods is presented in the table 2.

Table 1. Recognition ratios obtaind using an RDA classifier

Selection method (number of features)	Recognition ratio RDA	SVM-RDA
SFS (25)	55.3%	57.7%
Brute force (63)	56.1%	58.7%
CS-SFS (60)	59.2%	62.3%
CSO-SFS (41)	62.3%	64.2%

The accuracy measure used in this paper is the standard Q percentage accuracy (Baldi et al., [1]). Suppose there is $N = n_1 + n_2 + \ldots + n_p$ test proteins, where n_i is the number of proteins which belongs to the class i. Suppose that c_i of proteins from n_i are correctly recognised (as belonging to the class i). So the total number of $C = c_1 + c_2 + \ldots + c_p$ proteins is correctly recognized. Therefore the total accuracy is $Q = C/N$.

The results achieved using the proposed strategies are promising. The recognition rates obtained using these algorithms (57,7% - 64,2%) are better than those described in literature (51.2% - 61.2%) and better than obtained in our previous experiments (61.8%) [4]. The results are very encouraging. Our results

Table 2. Comparison among different methods

Method	Recognition ratio
SVM [6]	56.0%
HKNN [18]	57.4%
DIMLP-B [2]	61.2%
RS1_HKNN_K25 [17]	60.3%
RBFN [19]	51.2%
SVM-RDA [4]	61.8%
SVM-RDA (this paper)	64.2%

improved the recognition ratio achieved by other methods proposed in literature but however some extra experiments are needed. For example It will be interesting how our algorithm prove yourself for other problems with small number of samples.

References

1. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C., Nielsen, H.: Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16, 412–424 (2000)
2. Bologna, G., Appel, R.D.: A comparison study on protein fold recognition. In: Proceedings of the 9th ICONIP, Singapore, vol. 5, pp. 2492–2496 (2002)
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. Software (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
4. Chmielnicki, W., Stąpor, K.: Protein Fold Recognition with Combined SVM-RDA Classifier. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 162–169. Springer, Heidelberg (2010)
5. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice-Hall, Englewood Cliffs (1982)
6. Ding, C.H., Dubchak, I.: Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 17, 349–358 (2001)
7. Dubchak, I., Muchnik, I., Kim, S.H.: Protein folding class predictor for SCOP: approach based on global descriptors. In: Proceedings ISMB (1997)
8. Friedman, J.H.: Regularized Discriminant Analysis. Journal of the American Statistical Association 84(405), 165–175 (1989)
9. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, New York (1990)
10. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
11. Haindl, M., Somol, P., Ververidis, D., Kotropoulos, C.: Feature Selection Based on Mutual Correlation. In: Proceedings of Progress in Pattern Recognition, Image Analysis and Application, vol. 4225, pp. 569–577 (2006)
12. Hobohm, U., Sander, C.: Enlarged representative set of Proteins. Protein Sci. 3, 522–524 (1994)
13. Hobohm, U., Scharf, M., Schneider, R., Sander, C.: Selection of a representative set of structures from the Brookhaven Protein Bank. Protein Sci. 1, 409–417 (1992)
14. Lai, C., Reinders, M.J., Wessels, L.: Random subspace method for multivariate feature selection. Pattern Recognition Letters 27(10), 1067–1076 (2006)
15. Liu, C.L., Fujisawa, H.: Classification and Learning for Character Recognition: Comparison of Methods and Remaining Problems. In: Proc. Int. Workshop on Neural Networks and Learning in Document Analysis and Recognition, Seoul, Korea (2005)
16. Lo Conte, L., Ailey, B., Hubbard, T.J.P., Brenner, S.E., Murzin, A.G., Chotchia, C.: SCOP: A structural classification of protein database. Nucleic Acids Res. 28, 257–259 (2000)
17. Nanni, L.: A novel ensemble of classifiers for protein fold recognition. Neurocomputing 69, 2434–2437 (2006)
18. Okun, O.: Protein fold recognition with k-local hyperplane distance nearest neighbor algorithm. In: Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics, Pisa, Italy, September 24, pp. 51–57 (2004)
19. Pal, N.R., Chakraborty, D.: Some new features for protein fold recognition. In: Artificial Neural Networks and Neural Information Processing ICANN/ICONIP, Turkey, Istanbul, June 26–29, vol. 2714, pp. 1176–1183 (2003)
20. Shen, H.B., Chou, K.C.: Ensemble classifier for protein fold pattern recognition. Bioinformatics 22, 1717–1722 (2006)
21. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)

Empirical Comparison of Resampling Methods Using Genetic Neural Networks for a Regression Problem

Tadeusz Lasota[1], Zbigniew Telec[2], Grzegorz Trawiński[3], and Bogdan Trawiński[2]

[1] Wrocław University of Environmental and Life Sciences, Dept. of Spatial Management
ul. Norwida 25/27, 50-375 Wrocław, Poland
tadeusz.lasota@up.wroc.pl
[2] Wrocław University of Technology, Institute of Informatics,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{zbigniew.telec,bogdan.trawinski}@pwr.wroc.pl
[3] Wrocław University of Technology, Faculty of Electronics,
Wybrzeże S. Wyspiańskiego 27, 50-370 Wrocław, Poland
grzegorztrawinski@wp.pl

Abstract. In the paper the investigation of m-out-of-n bagging with and without replacement using genetic neural networks is presented. The study was conducted with a newly developed system in Matlab to generate and test hybrid and multiple models of computational intelligence using different resampling methods. All experiments were conducted with real-world data derived from a cadastral system and registry of real estate transactions. The performance of following methods was compared: classic bagging, out-of-bag, Efron's .632 correction, and repeated holdout. The overall result of our investigation was as follows: the bagging ensembles created using genetic neural networks revealed prediction accuracy not worse than the experts' method employed in reality.

Keywords: ensemble models, genetic neural networks, bagging, subagging.

1 Introduction

Bagging, which is one of the most effective computationally intensive procedures to improve unstable regressors and classifiers [23], has been focusing the attention of many researchers for last fifteen years. Bagging, which stands for **b**ootstrap **agg**regat**ing**, devised by Breiman [3] belongs to the most intuitive and simplest ensemble algorithms providing a good performance. Diversity of learners is obtained by using bootstrapped replicas of the training data. That is, different training data subsets are randomly drawn with replacement from the original base dataset. So obtained training data subsets, called also bags, are used then to train different classification or regression models. Finally, individual learners are combined through an algebraic expression, such as minimum, maximum, sum, mean, product, median, etc. [22]. The classic form of bagging is the *n-out-of-n with replacement* bootstrap where the number of samples in each bag equals to the cardinality of a base dataset and as a test set the whole original dataset is used. In order to achieve better computational effectiveness less overloading techniques were introduced which consisted in drawing from an original dataset smaller numbers of samples, with or

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 213–220, 2011.

without replacement. The *m-out-of-n without replacement* bagging, where at each step *m* observations less than *n* are distinctly chosen at random within the base dataset, belongs to such variants. This alternative aggregation scheme was called by Bühlmann and Yu [4] subagging for **sub**sample **agg**rega**ting**. In the literature the resampling methods of the same nature as subagging are also named Monte Carlo cross-validation [20] or repeated holdout [2]. In turn, subagging with replacement was called by Biau et al. [1] moon-bagging, standing for **m**-**o**ut-**o**f-**n** **b**ootstrap **agg**rega**ting**.

The statistical mechanisms of above mentioned resampling techniques are not yet fully understood and are still under active theoretical and experimental investigation [1], [2], [4], [5], [8], [9], [20]. Theoretical analyses and experimental results to date proved benefits of bagging especially in terms of stability improvement and variance reduction of learners for both classification and regression problems. Bagging techniques both with and without replacement may provide improvements in prediction accuracy in a range of settings. Moreover, n-out-of-n with replacement bootstrap and n/2-out-of-n without replacement sampling, i.e. half-sampling, may give fairly similar results.

The size of bootstrapped replicas in bagging usually is equal to the number of instances in an original dataset and the base dataset is commonly used as a test set for each generated component model. However, it is claimed it leads to an optimistic overestimation of the prediction error. So, as test error out-of-bag samples are applied, i.e. those included in the base dataset but not drawn to respective bags. These, in turn may cause a pessimistic underestimation of the prediction error. In consequence, the correction of the out-of-bag prediction error was proposed [2], [7].

The main focus of soft computing techniques to assist with real estate appraisals was directed towards neural networks [14], [19], [21], [24]. So far, we have investigated several methods to construct regression models to assist with real estate appraisal: evolutionary fuzzy systems, neural networks, decision trees, and statistical algorithms using MATLAB, KEEL, RapidMiner, and WEKA data mining systems [11], [15], [17]. We have studied also bagging ensemble models created applying these computational intelligence techniques [12], [16], [18].

The goal of the study presented in this paper was to compare m-out-of-n bagging with and without replacement with different sizes of samples with a property valuating method employed by professional appraisers in reality and a standard 10-fold cross validation. Genetic neural networks were applied to real-world regression problem of predicting the prices of residential premises based on historical data of sales/purchase transactions obtained from a cadastral system. The investigation was conducted with a newly developed system in Matlab to generate and test hybrid and multiple models of computational intelligence using different resampling methods.

2 Methods Used and Experimental Setup

The investigation was conducted with our new experimental system implemented in Matlab environment using Neural Network, Fuzzy Logic, Global Optimization, and Statistics toolboxes [6], [10]. The system was designed to carry out research into machine learning algorithms using various resampling methods and constructing and evaluating ensemble models for regression problems.

Real-world dataset used in experiments was drawn from a rough dataset containing above 50 000 records referring to residential premises transactions accomplished in one Polish big city with the population of 640 000 within 11 years from 1998 to 2008. The final dataset counted the 5213 samples for which the experts could estimate the value using their pairwise comparison method. Due to the fact that the prices of premises change substantially in the course of time, the whole 11-year dataset cannot be used to create data-driven models, therefore it was split into 20 half-year subsets. The sizes of half-year data subsets are given in Table 1.

Table 1. Number of instances in half-year datasets

1998-2	**1999-1**	**1999-2**	**2000-1**	**2000-2**	**2001-1**	**2001-2**	**2002-1**	**2002-2**	**2003-1**
202	213	264	162	167	228	235	267	263	267
2003-2	**2004-1**	**2004-2**	**2005-1**	**2005-2**	**2006-1**	**2006-2**	**2007-1**	**2007-2**	**2008-1**
386	278	268	244	336	300	377	289	286	181

In order to compare evolutionary machine learning algorithms with techniques applied to property valuation we asked experts to evaluate premises using their pairwise comparison method to historical data of sales/purchase transactions recorded in a cadastral system. The experts worked out a computer program which simulated their routine work and was able to estimate the experts' prices of a great number of premises automatically.

First of all the whole area of the city was divided into 6 quality zones. Next, the premises located in each zone were classified into 243 groups determined by 5 following quantitative features selected as the main price drivers: *Area, Year, Storeys, Rooms,* and *Centre.* Domains of each feature were split into three brackets as follows:

Area denotes the usable area of premises and comprises small flats up to 40 m^2, medium flats in the bracket 40 to 60 m^2, and big flats above 60 m^2.

Year (*Age*) means the year of a building construction and consists of old buildings constructed before 1945, medium age ones built in the period 1945 to 1960, and new buildings constructed between 1960 and 1996, the buildings falling into individual ranges are treated as in bad, medium, and good physical condition respectively.

Storeys are intended for the height of a building and are composed of low houses up to three storeys, multi-family houses from 4 to 5 storeys, and tower blocks above 5 storeys.

Rooms are designated for the number of rooms in a flat including a kitchen. The data contain small flats up to 2 rooms, medium flats in the bracket 3 to 4, and big flats above 4 rooms.

Centre stands for the distance from the city centre and includes buildings located near the centre i.e. up to 1.5 km, in a medium distance from the centre - in the brackets 1.5 to 5 km, and far from the centre - above 5 km.

Then the prices of premises were updated according to the trends of the value changes over time. Starting from the second half-year of 1998 the prices were updated for the last day of consecutive half-years. The trends were modelled by polynomials of degree three. Premises estimation procedure employed a two-year time window to take into consideration transaction data of similar premises.

1. Take 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 half-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.

In our study we employed an evolutionary approach to real-world regression problem of predicting the prices of residential premises based on historical data of sales/purchase transactions obtained from a cadastral system, namely genetic neural networks (GNN). Our GNN approach consisted in the evolution of connection weights with a predefined architecture of feedforward network with backpropagation comprising for five neurons in an input and hidden layers. The whole set of weights in a chromosome was represented by real numbers. Similar solutions are described in [13], [25]. Following resampling methods and their variants were applied in experiments and compared with the standard 10cv and the experts' method.

Classic: B100, B70, B50, B30 – m-out-of-n bagging with replacement with different sizes of samples using the whole base dataset as a test set. The numbers in the codes indicate what percentage of the base set was drawn to create training sets.

OoB: O100, O70, O50, O30 – m-out-of-n bagging with replacement with different sizes of samples tested with the out-of-bag (OoB) datasets. The numbers in the codes mean what percentage of the base dataset was drawn to create a training set.

Efron's .632: E100, E70, E50, E30 – models represent the Efron's 0.632 bootstrap method correcting the out-of-bag prediction error using the weighted average of the OoB error and so called resubstitution or apparent error with the weights equal to 0.632 and 0.368 respectively [2], [7].

k-Holdout: H90, H70, H50, H30 – m-out-of-n bagging without replacement with different sizes of samples. The numbers in the codes point out what percentage of the base dataset was drawn to create a training set.

In the case of bagging methods 50 bootstrap replicates (bags) were created on the basis of each base dataset, as performance functions the mean square error (MSE) was used, and as aggregation functions simple averages were employed. The normalization of data was accomplished using the min-max approach.

3 Results of Experiments

The performance of *Classic*, *OoB*, *Efron's .632*, and *k-Holdout* models created by genetic neural networks (GNN) in terms of MSE is illustrated graphically in Figures 1 and 2 respectively. In each figure for comparison the same results for *10cv* and *Expert* methods are shown. The Friedman test performed in respect of MSE values of all

models built over 20 half-year datasets showed that there are significant differences between some models. Average ranks of individual models are shown in Table 2, where the lower rank value the better model. In Table 3 and 4 the results of nonparametric Wilcoxon signed-rank test to pairwise comparison of the model performance are presented. The zero hypothesis stated there were not significant differences in accuracy, in terms of MSE, between given pairs of models. In both tables + denotes that the model in the row performed significantly better than, – significantly worse than, and ≈ statistically equivalent to the one in the corresponding column, respectively. In turn, / (slashes) separate the results for individual methods. The significance level considered for the null hypothesis rejection was 5%.

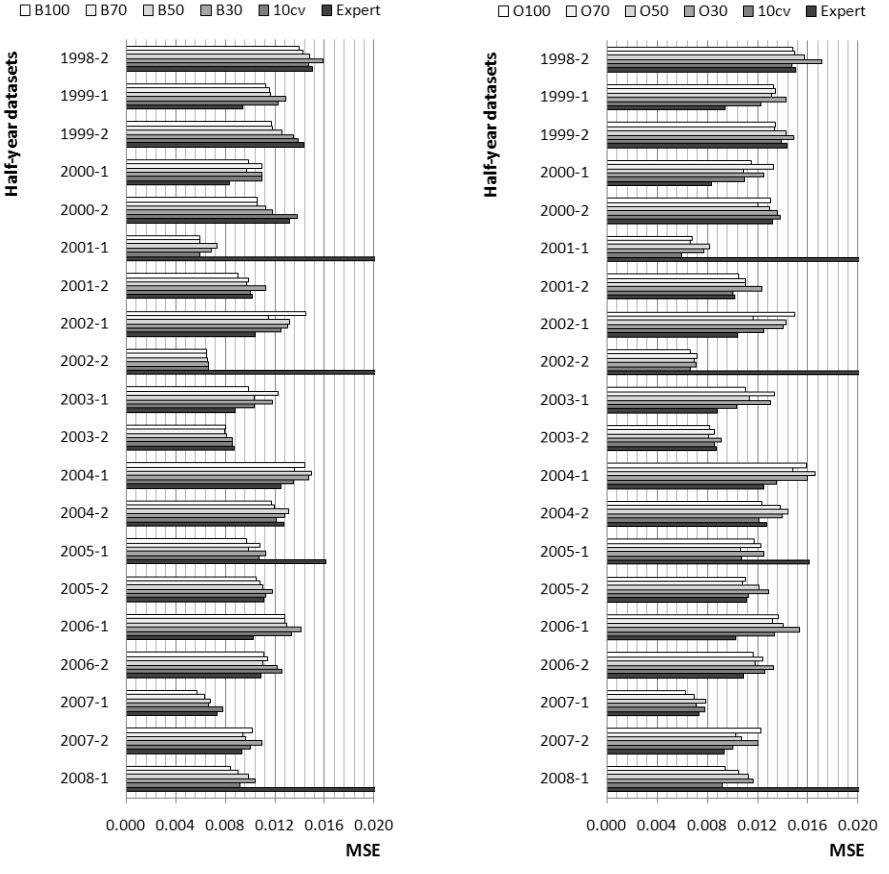


Fig. 1. Performance of *Classic* (left) and *OoB* (right) models generated using GNN

Table 2. Average rank positions of models determined during Friedman test

Method	1st	2nd	3rd	4th	5th	6th
Classic	B100 (2.25)	B70 (2.50)	B50 (3.60)	Expert (3.60)	10cv (4.05)	B30 (5.00)
OoB	10cv (2.50)	O100 (3.00)	Expert (3.10)	O70 (3.25)	O50 (4.00)	O30 (5.15)
Efron's .632	E100 (2.70)	E70 (3.20)	Expert (3.40)	E50 (3.50)	10cv (3.50)	E30 (4.70)
k-Holdout	10cv (2.60)	H90 (2.70)	Expert (3.00)	H70 (3.70)	H50 (3.95)	H30 (5.05)

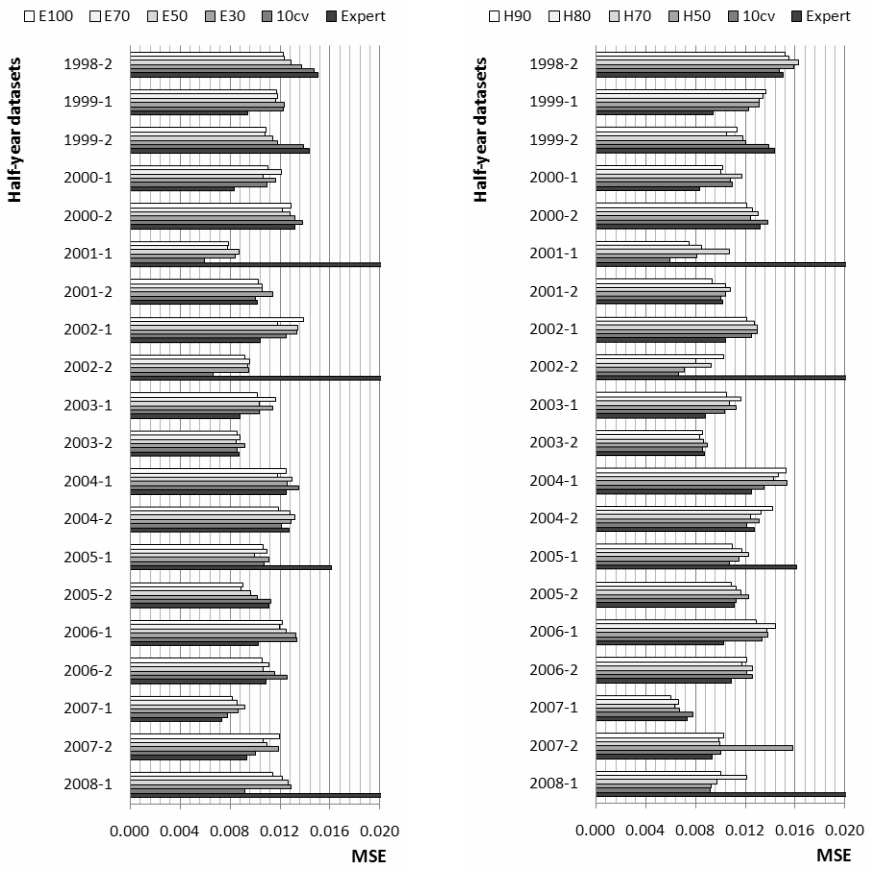


Fig. 2. Performance of *Efron's .632* (left) and *k-Holdout* (right) models created by GNN

Table 3. Results of Wilcoxon tests for the performance of bagging with replacement models

	B/O/E100	B/O/E70	B/O/E50	B/O/E30	10cv	Expert
B/O/E100		≈ / ≈ / ≈	+ / ≈ / ≈	+ / + / +	+ / ≈ / ≈	≈ / ≈ / ≈
B/O/E70	≈ / ≈ / ≈		≈ / ≈ / ≈	+ / + / +	+ / ≈ / ≈	≈ / ≈ / ≈
B/O/E50	– / ≈ / ≈	≈ / ≈ / ≈		+ / + / +	≈ / – / ≈	≈ / ≈ / ≈
B/O/E30	– / – / –	– / – / –	– / – / –		– / – / ≈	≈ / ≈ / ≈
10cv	– / ≈ / ≈	– / ≈ / ≈	≈ / + / ≈	+ / + / ≈		≈ / ≈ / ≈
Expert	≈ / ≈ / ≈	≈ / ≈ / ≈	≈ / ≈ / ≈	≈ / ≈ / ≈	≈ / ≈ / ≈	

Table 4. Results of Wilcoxon tests for the performance of bagging without replacement models

	H90	H70	H50	H30	10cv	Expert
H90		≈	≈	+	≈	≈
H70	≈		≈	+	–	≈
H50	≈	≈		+	≈	≈
H30	–	–	–		–	≈
10cv	≈	+	≈	+		≈
Expert	≈	≈	≈	≈	≈	

The general outcome is as follows: the performance of the experts' method fluctuated strongly achieving for some datasets excessively high MSE values and for others the lowest values; MSE ranged from 0.007 to 0.023. The models created over 30% subsamples performed significantly worse than ones trained using bigger portions of base datasets for all methods. More specifically, no significant differences between B100 and B70 were observed. B100 and B70 provided better results than 10cv. No significant differences were noticed among O100, O70, and 10cv. In turn, 10cv turned out to be better than O50. No significant differences among E100, E70, E50, and 10cv were seen. H90 and H50 did not show any significant difference when compared to H70 and 10cv.

A separate Wilcoxon test showed that B100 performed significantly better than H50. Thus, our tests did not confirm the observation presented in the literature that classic bagging and half-sampling provide statistically equivalent results.

4 Conclusions and Future Work

The experiments, aimed to compare the performance of bagging ensembles built using genetic neural networks over real-world data taken from a cadastral system with different numbers of training samples drawn from the base dataset with and without replacement. The performance of following methods was compared: three variants of m-out-of-n bagging with replacement, i.e. classic bagging, out-of-bag, Efron's .632 correction, and one m-out-of-n bagging without replacement called also subagging or repeated holdout. Moreover, the predictive accuracy of a pairwise comparison method applied by professional appraisers in reality was compared with soft computing machine learning models for residential premises valuation.

The overall results of our investigation were as follows. The bagging ensembles created using genetic neural networks revealed prediction accuracy not worse than the experts' method employed in reality. It confirms that automated valuation models can be successfully utilized to support appraisers' work.

We plan to continue exploring resampling methods ensuring faster data processing such as random subspaces, subsampling, and techniques of determining the optimal sizes of multi-model solutions. This can lead to achieve both low prediction error and an appropriate balance between accuracy, complexity, stability.

Acknowledgments. This paper was partially supported by the Polish National Science Centre under grant no. N N516 483840.

References

1. Biau, G., Cérou, F., Guyader, A.: On the Rate of Convergence of the Bagged Nearest Neighbor Estimate. Journal of Machine Learning Research 11, 687–712 (2010)
2. Borra, S., Di Ciaccio, A.: Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. Computational Statistics & Data Analysis 12, 2976–2989 (2010)
3. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)
4. Bühlmann, P., Yu, B.: Analyzing bagging. Annals of Statistics 30, 927–961 (2002)
5. Buja, A., Stuetzle, W.: Observations on bagging. Statistica Sinica 16, 323–352 (2006)

6. Czuczwara, K.: Comparative analysis of selected evolutionary algorithms for optimization of neural network architectures. Master's Thesis (in Polish), Wrocław University of Technology, Wrocław, Poland (2010)
7. Efron, B., Tibshirani, R.J.: Improvements on cross-validation: the 632+ bootstrap method. Journal of the American Statistical Association 92(438), 548–560 (1997)
8. Friedman, J.H., Hall, P.: On bagging and nonlinear estimation. Journal of Statistical Planning and Inference 137(3), 669–683 (2007)
9. Fumera, G., Roli, F., Serrau, A.: A theoretical analysis of bagging as a linear combination of classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(7), 1293–1299 (2008)
10. Góral, M.: Comparative analysis of selected evolutionary algorithms for optimization of fuzzy models for real estate appraisals. Master's Thesis (in Polish), Wrocław University of Technology, Wrocław, Poland (2010)
11. Graczyk, M., Lasota, T., Trawiński, B.: Comparative Analysis of Premises Valuation Models Using KEEL, RapidMiner, and WEKA. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 800–812. Springer, Heidelberg (2009)
12. Graczyk, M., Lasota, T., Trawiński, B., Trawiński, K.: Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) Intelligent Information and Database Systems. LNCS (LNAI), vol. 5991, pp. 340–350. Springer, Heidelberg (2010)
13. Kim, D., Kim, H., Chung, D.: A Modified Genetic Algorithm for Fast Training Neural Networks. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3496, pp. 660–665. Springer, Heidelberg (2005)
14. Kontrimas, V., Verikas, A.: The mass appraisal of the real estate by computational intelligence. Applied Soft Computing 11(1), 443–448 (2011)
15. 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)
16. Krzystanek, M., Lasota, T., Telec, Z., Trawiński, B.: Analysis of Bagging Ensembles of Fuzzy Models for Premises Valuation. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) Intelligent Information and Database Systems. LNCS (LNAI), vol. 5991, pp. 330–339. Springer, Heidelberg (2010)
17. Lasota, T., Mazurkiewicz, J., Trawiński, B., Trawiński, K.: Comparison of Data Driven Models for the Validation of Residential Premises using KEEL. International Journal of Hybrid Intelligent Systems 7(1), 3–16 (2010)
18. Lasota, T., Telec, Z., Trawiński, B., Trawiński, K.: Exploration of Bagging Ensembles Comprising Genetic Fuzzy Models to Assist with Real Estate Appraisals. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 554–561. Springer, Heidelberg (2009)
19. Lewis, O.M., Ware, J.A., Jenkins, D.: A novel neural network technique for the valuation of residential property. Neural Computing & Applications 5(4), 224–229 (1997)
20. Molinaro, A.N., Simon, R., Pfeiffer, R.M.: Prediction error estimation: A comparison of resampling methods. Bioinformatics 21(15), 3301–3307 (2005)
21. Peterson, S., Flangan, A.B.: Neural Network Hedonic Pricing Models in Mass Real Estate Appraisal. Journal of Real Estate Research 31(2), 147–164 (2009)
22. Polikar, R.: Ensemble Learning. Scholarpedia 4(1), 2776 (2009)
23. Schapire, R.E.: The Strength of Weak Learnability. Mach. Learning 5(2), 197–227 (1990)
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(2), 185–201 (1995)
25. Yao, X.: Evolving artificial neural networks. Proceedings of the IEEE 87(9), 1423–1444 (1999)

Structured Output Element Ordering in Boosting-Based Classification

Tomasz Kajdanowicz and Przemyslaw Kazienko

Wroclaw University of Technology, Wroclaw, Poland,
Faculty of Computer Science and Management
{tomasz.kajdanowicz,kazienko}@pwr.wroc.pl

Abstract. The paper describes the method for structured output classification that is able to perform classification task for unknown shape of output structure. In previous work authors provided that the classification of the element in the output structure can be performed using standard input of the instance (its profile) as well as all other preceding output elements (already classified) as learning attributes. Now they present how the order of the score function application to each element of output space is important and may determine the overall accuracy. For that reason the paper addresses the crucial problem of how to order elements in the structured learning process to get greater final accuracy. The learning is performed by means of ensemble, boosting classification method adapted to structured prediction - the AdaBoostSeq algorithm according to several ordering methods. A heuristic method of ordering is worked out as well. Experimental studies were carried out on a number of real financial datasets.

Keywords: Structured Output Learning, Structured Output Prediction, Classifier Fusion, Ensembles, Multiple Classifier Systems.

1 Introduction

Recently, it has been observed a growing interest in machine learning algorithms which are able to operate on structured data. Such a data consists of a set of training input-output pairs but either the input or the output (sometimes both of them) are more complex in comparison with traditional data types, namely vectors. Structured prediction may deal with the output space that contains a complex structure like sequences, trees or graphs. However, the combined applicability and generality of learning in complex spaces result with a number of significant theoretical and practical challenges.

Standard classification or regression problems tend to discover the mapping to output space that is either real-valued or a value from a small, unstructured set of labels. Structured output prediction, in opposite, deals with the nature of output space that might contain a structure rich in information that, in turn, can be utilized in the learning process. Due to the profile of structured output prediction, it requires in some cases an optimization or search mechanism over complete output space, that makes the problem nontrivial.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 221–228, 2011.

In the community of hybrid information systems there exists a variety of structured output prediction problems [2], for instance protein function classification, semantic classification of images or text categorization. The structured output prediction (classification) methods solving such problems might be designed on the basis of two distinct approaches. The former utilizes *a priori* known information about the structure, i.e. makes use of sequential structure, i.e. the algorithm tries to benefit from relations between predecessors and consequents. The latter does not assume any shape of output structure and attempts to perform prediction on the pure data.

2 Related Work

The most obvious and directly arising method for structured output prediction is a probabilistic model jointly considering the input and output variables. There exist many examples of such probabilistic models for a variety of inputs and outputs, e.g. probabilistic graphical models or stochastic context free grammars. Given an input, the predicted output might be determined as the result of maximization of the posteriori probability, namely the technique of maximum posteriori estimation. In such approaches, the learning is to model the joint input and output data distribution. However, it is well known that this approach of first modeling the distribution and subsequently using maximum a posteriori estimation for prediction is indirect and might be suboptimal. Therefore, the other, direct discriminative approach might be more appropriate. Such a discriminative learning algorithms perform a prediction on the basis of scoring function optimization over the output space. Recently presented and studied discriminative learning algorithms include Max-margin Markov Nets that consider the structured output prediction problem as a quadratic programming problem [10], a bit more flexible approach that is an alternative adjustment of logistic regression to the structured outputs called Conditional Random Fields [8], Structured Perceptron [1] that has minimal requirements on the output space shape and is easy to implement, Support Vector Machine for Interdependent and Structured Outputs (SVM^{STRUCT}) [12], which applies variety of loss functions and an example of adaptation of the popular backpropagation algorithm - BPMLL [14] where a new error function takes multiple targets into account. Another example of structured output algorithm is an extention of the original AdaBoost algorithm to the sequence labeling case without reducing it to several simple two-class problems; it is the AdaBoostSeq algorithm proposed by authors in [6] and utilized in this paper within experimental studies.

Summarizing, both presented approaches, the generative modeling and the discriminative learning algorithms make use of scoring function to score each element of the output space. In case of iterative algorithms like AdaBoostSeq [6] the order of the score function application to each element of output space is important and may determine the overall accuracy. Therefore appropriate output space learning order is required to provide best possible generalization.

3 Structured Output Learning Using AdaBoostSeq Algorithm and Heuristics

It was proposed the modification of the cost function as well as the new structure of sequential increments in AdaBoostSeq algorithm. It was originally based on the most popular boosting algorithm, AdaBoost [3,11,13].

It is assumed that the structured output classification is a binary sequence classification problem with $y_i^\mu \in \{-1, 1\}$, for $i = 1, 2, \ldots, N$ and $\mu = 1, 2, \ldots, T$, where N is the number of cases, T is the length of the sequence, i.e. the number of elements in the structure. The general goal of the multiple model method is to construct T optimally designed linear combinations of K base classifiers:

$$\forall \mu = 1, 2, \ldots, T \quad F^\mu(x) = \sum_{k=1}^{K} \alpha_k^\mu \Phi(x, \Theta_k^\mu) \tag{1}$$

where: $F^\mu(x)$ is the combined, final meta classifier for the μth sequence item (structure element); $\Phi(x, \Theta_k^\mu)$ represents the kth base classifier, performing according to its Θ_k^μ parameters and returning a binary class label for each instance (case) x; α_k^μ is the weight associated to the kth classifier.

Values of the unknown parameters (α_k^μ and Θ_k^μ) result from minimization of prediction error for each μth sequence element for all K classifiers obtained in stage-wise suboptimal method in M steps [11]. The optimized sequence-loss balancing cost function J is defined as:

$$J(\alpha^\mu, \Theta^\mu) = \sum_{i=1}^{N} \exp(-y_i^\mu(\xi F_{m-1}^\mu(x_i) + (1-\xi) y_i^\mu \hat{R}_m^\mu(x_i) + \alpha^\mu \Phi(x_i, \Theta^\mu))) \tag{2}$$

where: $\hat{R}_m^\mu(x_i)$ is an impact function denoting the influence on prediction according to the quality of preceding sequence labels predictions; ξ is the parameter that allows controlling the influence of impact function in weights composition, $\xi \in \langle 0, 1 \rangle$.

$\hat{R}_m^\mu(x_i)$ is applied in computation for current sequence position, as follows:

$$\hat{R}_m^\mu(x_i) = \sum_{j=1}^{m-1} \alpha_j^\mu R^\mu(x_i) \tag{3}$$

$$R^\mu(x_i) = \frac{\sum_{l=1}^{\mu} y_i^l \frac{F^l(x_i)}{\sum_{k=1}^{K} \alpha_k^l}}{\mu} \tag{4}$$

where: $R^\mu(x_i)$ is the auxiliary function that denotes the average coincidence between prediction result $F^l(x_i)$ and the actual value y_i^l weighted with the weights α_k^l associated to the kth base classifiers for all sequence items achieved so far (from 1 to μ) with respect to the value of μ.

The impact function $\hat{R}_m^{\mu}(x_i)$, introduced in Eq. 3 and 4, measures the correctness of prediction for all preceding labels $l = 1, \ldots, \mu$ in the sequence. This function is utilized in the cost function and it provides the smaller error deviation for the whole sequence. The greater compliance between prediction and the real value, the higher the function value.

The presented above brief explanation lead to the AdaBoostSeq algorithm for sequence prediction and may be found as full version in [7]. Overall, the AdaBoostSeq algorithm performs the learning of structured output in sequential manner, step by step for each of the structured output elements and results obtained in the previous steps are utilized as an additional input for the following elements. Therefore, the order of elements (the order of learning) may have significant impact on the overall classification accuracy.

As the complete overview over the space of all possible ordering solutions is exponentially complex there should be a method providing reasonable ordering in acceptable time. In order to provide the ordering of the learning steps a new approach formalized as a heuristics in the searching process is proposed.

The proposed heuristics is constructed on the basis of classification error minimization. The heuristics determine which output element should be taken in the next step of learning process. It is done by quick assessment of the classification error for all remaining elements in the structured output. The element that is classified with the smallest error is taken as next in ordering. Therefore in the structured output of n elements it is needed to perform $\frac{n^2+n}{2} - 1$ evaluations to propose the ordering. The evaluation contains the learning and testing phase. For instance if we consider structured output classification of 10 elements the heuristics will require 10 error evaluations in order to determine the first element in the learning. The second element will be chosen from 9 and so on, up to the tenth element, together 54 evaluations.

4 Experiments and Results

The main objective of performed experiments was to evaluate the classification accuracy of the AdaBoostSeq algorithm driven by various learning ordering of output elements. The method was examined according to hamming loss and classification accuracy for six distinct datasets. The utilized measures are calculated based on the differences of the actual and the predicted sets of labels (classes) over all cases x_i in the test set. The first measure is Hamming Loss HL, which was proposed in [9] and is defined as:

$$HL = \frac{1}{N} \sum_{i=1}^{N} \frac{Y_i \triangle F(x_i)}{|Y_i|} \tag{5}$$

where: N is the total number of cases x_i in the test set; Y_i denotes actual (real) labels (classes) in the sequence, i.e. entire structure corresponding to instance x_i; $F(x_i)$ is a sequence of labels predicted by classifier and $\triangle$ stands for the symmetric difference of two vectors, which is the vector-theoretic equivalent of the exclusive disjunction in Boolean logic.

Fig. 1. Average relative Hamming Loss (HL) error in comparison with result obtained with the best ordering (1- Sequential ordering, 2 - Reverse sequential ordering, 3 - Best ordering , 4 - Heuristic ordering, 5 - Worst ordering, 6 - Random ordering)

The second evaluation measure utilized in the experiments is Classification Accuracy CA [4], defined as follows:

$$CA = \frac{1}{N}\sum_{i=1}^{N} I(Y_i = F(x_i)) \tag{6}$$

where: $I(true) = 1$ and $I(false) = 0$.

Measure CA is a very strict evaluation measure as it requires the predicted sequence of labels to be an exact match of the true set of labels.

The main attention in the experiment was concentrated on the evaluation of six distinct ordering schemes that have been implemented. These ordering schemes were proposed to examine weather the order of output elements being learned is really important and if so, how to propose reasonable heuristic ordering method that may be efficient and provide ability to result with high accuracy.

Six distinct ordering schemes were examined:

1. Sequential ordering (from left to right)
2. Reverse sequential ordering (from right to left)
3. Best ordering (with the best classification accuracy among all others)
4. Heuristic ordering (ordering that was obtained by the procedure explained in Sec. 3)
5. Worst ordering (ordering that provides the worst classification accuracy)
6. Random ordering (obtained randomly from uniform distribution)

The performance of analyzed methods was evaluated using 10-fold cross-validation and the evaluation measures from Eq. 5 and Eq. 6. These two metrics are widely-used in the literature and are indicative for the performance.

The experiments were carried out on six distinct, real datasets from the same application domain of debt portfolio pattern recognition [5]. Datasets represent the problem of aggregated prediction of sequential repayment values over time for a set of claims. The structured output for each debt (case) is composed of a vector of binary indicators denoting whether in given period of time it was repaid at a

certain level. The output to be predicted is provided for all consecutive periods of time the case was repaid. For the purpose of the experiment, the output was limited to only six elements. The number of cases in the datasets varied from 1,703 to 6,818, while the number of initial, numeric input attributes was the same: 25. Note that the number of input attributes refers only classification for the first element in the sequence; for the others, outputs of the preceding elements are added to their input.

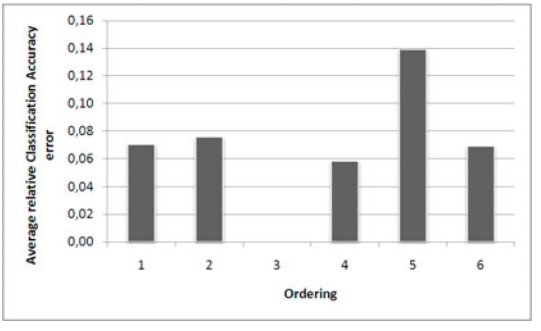


Fig. 2. Average relative Classification Accuracy error in comparison with result obtained with the best ordering (1- Sequential ordering, 2 - Reverse sequential ordering, 3 - Best ordering , 4 - Heuristic ordering, 5 - Worst ordering, 6 - Random ordering)

Table 1. Results obtained in the experiments, where orderings denote: 1- Sequential ordering, 2 - Reverse sequential ordering, 3 - Best ordering , 4 - Heuristic ordering, 5 - Worst ordering, 6 - Random ordering; HL - Hamming Loss, CA - Classification Accuracy

	Ordering schema											
	1		2		3		4		5		6	
Dataset	HL	CA	HL	CA	HL	CA	HL	CA	HL	CA	HL	CA
1	0.056	0.764	0.056	0.769	0.046	0.801	0.051	0.772	0.063	0.730	0.054	0.767
2	0.090	0.689	0.099	0.692	0.083	0.725	0.097	0.680	0.109	0.639	0.097	0.686
3	0.091	0.726	0.090	0.765	0.080	0.776	0.091	0.733	0.113	0.693	0.096	0.734
4	0.140	0.551	0.146	0.543	0.123	0.607	0.135	0.576	0.153	0.519	0.139	0.562
5	0.131	0.467	0.137	0.441	0.121	0.500	0.135	0.451	0.142	0.432	0.133	0.461
6	0.146	0.525	0.146	0.507	0.124	0.583	0.147	0.556	0.172	0.444	0.151	0.519
Average	0.109	0.620	0.112	0.620	0.096	0.665	0.109	0.628	0.125	0.576	0.112	0.621

The results with values of Hamming Loss and Classification Accuracy are presented in Tab. 1. Its relative values, with respect to the best possible order (no. 3), i.e. $(x - best)/best \cdot 100\%$, are depicted in Tab. 2, Fig. 1, and Fig. 4.

As we can see, the average differences between the worst (no. 5) and the best results (no. 3) are about 30% (for HL) and 14% (for CA). This is the margin, within which some heuristic methods can be searched. The regular order (no. 1)

Table 2. Average relative errors in comparison with the best ordering, in percentage (0 denotes the same accuracy like obtained with the best ordering), the orderings are: 1- Sequential ordering, 2 - Reverse sequential ordering, 3 - Best ordering , 4 - Heuristic ordering, 5 - Worst ordering, 6 - Random ordering; HL - Hamming Loss, CA - Classification Accuracy

	Ordering schema											
	1		2		3		4		5		6	
Dataset	HL	CA	HL	CA	HL	CA	HL	CA	HL	CA	HL	CA
1	22.69	4.59	22.90	3.93	0.00	0.00	12.61	3.54	39.08	8.78	19.04	4.25
2	8.77	5.04	18.64	4.56	0.00	0.00	16.95	6.27	31.13	11.93	16.28	5.47
3	13.92	6.44	11.92	1.38	0.00	0.00	12.80	5.47	40.62	10.68	19.63	5.34
4	13.54	9.28	18.83	10.61	0.00	0.00	10.13	5.15	24.73	14.59	13.26	7.38
5	7.60	6.64	12.79	11.74	0.00	0.00	11.14	9.78	16.98	13.59	9.30	7.86
6	17.81	10.06	17.09	13.10	0.00	0.00	18.13	4.69	38.06	23.83	21.47	11.04
Average	14.05	7.01	17.03	7.55	0.00	0.00	13.63	5.82	31.76	13.90	16.50	6.89

is somewhere in the middle between the best and the worst one. The reverse order of elements (no. 2) are always worse than the regular one (no. 1). An average random order (no. 6) is better than the reverse order (no. 2) and either better (for CA) or worse (for HL) than the regular order. The heuristic method of ordering (no. 4) is usually better than other ordering methods (no. 1, 2, 5, 6). In particular, this heuristic approach usually gains in both CA and HL compared to the regular order. The heuristic method appears to be a rational solution between the simple orders (like regular, reverse or random ones) and searching for the best one, what is computationally too expensive.

5 Conclusions

The problem of structured output elements ordering in the prediction was considered in the paper. The studied problem was implemented according to AdaBoostSeq algorithm based on the assumption that labels of the already classified output elements are used as additional input features for the next elements. Since the elements in the sequence may be correlated, their order of learning may influence accuracy of the entire classification. According to experiments, the margin between the worst and the best order may be even several dozen of percent for hamming loss measure and a dozen for classification accuracy. Moreover, the natural, sequential order, did not provide the best possible classification results even if the nature of studied problem would indicate that. The heuristic order proposed in the paper appears to be a reasonable direction to find the order which provides better results than other simple orders; it is simultaneously computationally much less expensive than checking all possible orders to find the best one.

Further studies will focus on development of some other heuristic ordering methods and experiments on other datasets.

Acknowledgement

This work was supported by The Polish Ministry of Science and Higher Education the research project 2011-2014, and Fellowship co-financed by European Union within European Social Fund.

References

1. Collins, M.: Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In: Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 1–8 (2002)
2. Corchado, E., Abraham, A., Ferreira de Carvalho, A.: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)
3. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139 (1997)
4. Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: Proceedings of the ACM Conference on Information and Knowledge Management, pp. 195–200 (2005)
5. Kajdanowicz, T., Kazienko, P.: Prediction of Sequential Values for Debt Recovery. In: Bayro-Corrochano, E., Eklundh, J.-O. (eds.) CIARP 2009. LNCS, vol. 5856, pp. 337–344. Springer, Heidelberg (2009)
6. Kajdanowicz, T., Kazienko, P.: Boosting-based Sequence Prediction. New Generation Computing (2011)
7. Kazienko, P., Kajdanowicz, T.: Base Classifiers in Boosting-based Classification of Sequential Structures. Neural Network World 20, 839–851 (2010)
8. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning ICML, pp. 282–289 (2001)
9. Schapire, R.E., Singer, Y.: Boostexter: a boosting-based system for text categorization. Machine Learning 39, 135–168 (2000)
10. Taskar, B., Guestrin, C., Koller, D.: Max-margin Markov networks. In: Advances in Neural Information Processing Systems, vol. 16, pp. 25–32. MIT Press, Cambridge (2004)
11. Theodoris, S., Koutroumbas, K.: Pattern Recognition. Elsevier, Amsterdam (2009)
12. Tsochantaridis, I., Hofmann, T., Thorsten, J., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research 6, 1453–1484 (2005)
13. Wozniak, M.: Proposition of Boosting Algorithm for Probabilistic Decision Support System. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2004. LNCS, vol. 3036, pp. 675–678. Springer, Heidelberg (2004)
14. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering 18, 1338–1351 (2006)

Probabilistic Approach to the Dynamic Ensemble Selection Using Measures of Competence and Diversity of Base Classifiers

Rafal Lysiak, Marek Kurzynski, and Tomasz Woloszynski

Wroclaw University of Technology, Dept. of Systems and Computer Networks,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
{rafal.lysiak,marek.kurzynski,tomasz.woloszynski}@pwr.wroc.pl

Abstract. In the paper measures of classifier competence and diversity using a probabilistic model are proposed. The multiple classifier system (MCS) based on dynamic ensemble selection scheme was constructed using both measures developed. The performance of proposed MCS was compared against three multiple classifier systems using six databases taken from the UCI Machine Learning Repository and the StatLib statistical dataset. The experimental results clearly show the effectiveness of the proposed dynamic selection methods regardless of the ensemble type used (homogeneous or heterogeneous).

Keywords: Dynamic ensemble selection, Classifier competence, Diversity measure.

1 Introduction

Dynamic ensemble selection (DES) methods are recently intensively developed as an effective approach to the construction of multiple classifier systems ([10], [14], [17]). In these methods, first an ensemble of base classifiers is dynamically selected and then the selected classifiers are combined by majority voting. The most DES schemes use the concept of classifier competence on a defined neighbourhood or region [15], such as the local accuracy estimation [19], [6], [16], Bayes confidence measure [12], multiple classifier behaviour [11] or probabilistic model [17], among others.

The performance of multiclassifier system based on DES approach can be significantly improved through the use of diverse ensemble of classifiers ([2], [4], [5]). Then, it is intuitively clear that classifiers to be selected should be competent (accurate) as well as diverse (different from one another).

In this paper a novel procedure for ensemble selection is developed, in which both a competence and a diversity of member classifiers are simultaneously taken into consideration in the dynamic fashion. Methods for calculating classifier competence and diversity using a probabilistic model are based on the original concept of a randomized reference classifier (RRC) [17], which – on average – acts like classifier evaluated. The competence of a classifier is calculated as the probability of correct classification of the respective RRC and the class-dependent

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 229–236, 2011.

error probabilities of RRC are used for determining the diversity measure, which evaluates the difference of incorrect outputs of classifiers [1]. Next, the procedure for dynamic ensemble selection using both measures is proposed, in which incompetent classifiers are eliminated and the ensemble is kept maximally diverse. The subset of classifiers selected in the DES procedure is combined using continuous-valued outputs and weighted majority voting where the weights are equal to the competence values.

The paper is organized as follows. In section 2 the randomized reference classifier (RRC) is presented and measures of base classifier competence and ensemble diversity are developed. Section 3 describes the multiple classifier system that was constructed using both measures. The experiments conducted and results with discussion are presented in section 4. Section 5 concludes the paper.

2 Theoretical Framework

2.1 Preliminaries

Consider a classification problem with a set $\mathcal{M} = \{1, 2, \ldots, M\}$ of class labels and a feature space $\mathcal{X} \subseteq \mathcal{R}^n$. Let a pool of classifiers, i.e. a set of trained classifiers $\Psi = \{\psi_1, \psi_2, \ldots, \psi_L\}$ be given. Let

$$\psi_l : \mathcal{X} \to \mathcal{M} \tag{1}$$

be a classifier, that produces a vector of discriminant functions $[d_{l1}(x), d_{l2}(x), \ldots, d_{lM}(x)]$ for an object described by a feature vector $x \in \mathcal{X}$. The value of $d_{lj}(x)$, $j \in \mathcal{M}$ represents a support given by the classifier ψ_l for the fact that the object x belongs to the j-th class. Assume without loss of generality that $d_{lj}(x) \geq 0$ and $\sum_j d_{lj}(x) = 1$. Classification is made according to the maximum rule

$$\psi_l(x) = i \Leftrightarrow d_{li}(x) = \max_{j \in \mathcal{M}} d_{lj}(x). \tag{2}$$

Now, our purpose is to determine the following characteristics, which will be the basis for dynamic selection of classifiers from the pool:

1. a competence measure $C(\psi_l|x)$ of each base classifier ($l = 1, 2, \ldots, L$), which evaluates the competence of classifier ψ_l i.e. its capability to correct activity (correct classification) at a point $x \in \mathcal{X}$.
2. a diversity measure $D(\Psi_E|x)$ of any ensemble of base classifiers Ψ_E, considered as the independency of the errors made by the member classifiers at a point $x \in \mathcal{X}$.

In this paper trainable competence and diversity functions are proposed using a probabilistic model. It is assumed that a learning set

$$\mathcal{S} = \{(x_1, j_1), (x_2, j_2), \ldots, (x_N, j_N)\}; \quad x_k \in \mathcal{X}, \ j_k \in \mathcal{M} \tag{3}$$

is available for the training of competence and diversity measures.

In the next section the original concept of a reference classifier will be presented, which – using probabilistic model – will state the convenient and effective tool for determining both competence and diversity measures.

2.2 Randomized Reference Classifier - RRC

A classifier[1] ψ from the pool Ψ is modeled by a randomized reference classifier (RRC) [17] which takes decisions in a random manner. A randomized decision rule (classifier) is, for each $x \in \mathcal{X}$, a probability distribution on a decision space [3] or – for the classification problem (2) – on the product $[0,1]^M$, i.e. the space of vectors of discriminant functions (supports).

The RRC classifies object $x \in \mathcal{X}$ according to the maximum rule (2) and it is constructed using a vector of class supports $[\delta_1(x), \delta_2(x), \ldots, \delta_M(x)]$ which are observed values of random variables (rvs) $[\Delta_1(x), \Delta_2(x), \ldots, \Delta_M(x)]$. Probability distributions of the random variables satisfy the following conditions:

(1) $\Delta_j(x) \in [0,1]$;
(2) $E[\Delta_j(x)] = d_j(x),\ j = 1, 2, \ldots, M$;
(3) $\sum_{j=1,2,\ldots,M} \Delta_j(x) = 1$,

where E is the expected value operator. In other words, class supports produced by the modeled classifier ψ are equal to the expected values of class supports produced by the RRC.

Since the RRC performs classification in a stochastic manner, it is possible to calculate the probability of classification an object x to the i-th class:

$$P^{(RRC)}(i|x) = Pr[\forall_{k=1,\ldots,M,\ k\neq i}\ \Delta_i(x) > \Delta_k(x)]. \tag{4}$$

In particular, if the object x belongs to the i-th class, from (4) we simply get the conditional probability of correct classification $Pc^{(RRC)}(x)$.

The key element in the modeling presented above is the choice of probability distributions for the rvs $\Delta_j(x), j \in \mathcal{M}$ so that the conditions 1-3 are satisfied. In this paper beta probability distributions are used with the parameters $\alpha_j(x)$ and $\beta_j(x)$ $(j \in \mathcal{M})$. The justification of the choice of the beta distribution can be found in [17] and furthermore the MATLAB code for calculating probabilities (4) was developed and it is freely available for download [18].

Applying the RRC to a learning point x_k and putting in (4) $i = j_k$, we get the probability of correct classification of RRC at a point $x_k \in \mathcal{S}$, namely

$$Pc^{(RRC)}(x_k) = P^{(RRC)}(j_k|x_k), \quad x_k \in \mathcal{S}. \tag{5}$$

Similarly, putting in (4) a class $j \neq j_k$ we get the class-dependent error probability at a point $x_k \in \mathcal{S}$:

$$Pe^{(RRC)}(j|x_k) = P^{(RRC)}(j|x_k), \quad x_k \in \mathcal{S}, \quad j(\neq j_k) \in \mathcal{M}. \tag{6}$$

In next sections probabilities of correct classification (5) and conditional probabilities of error (6) for learning objects will be utilized for determining the competence and diversity functions of base classifiers.

[1] Throughout this subsection, the index l of classifier ψ_l and class suppports $d_{lj}(x)$ is omitted for clarity.

2.3 Measure of Classifier Competence

Since the RRC can be considered equivalent to the modeled base classifier $\psi_l \in \Psi$, it is justified to use the probability (5) as the competence of the classifier ψ_l at the learning point $x_k \in \mathcal{S}$, i.e.

$$C(\psi_l|x_k) = Pc^{(RRC)}(x_k). \tag{7}$$

The competence values for the validation objects $x_k \in \mathcal{S}$ can be then extended to the entire feature space $\mathcal{X}$. To this purpose the following normalized Gaussian potential function model was used ([17]):

$$C(\psi_l|x) = \frac{\sum_{x_k \in \mathcal{S}} C(\psi_l|x_k) exp(-dist(x, x_k)^2)}{\sum_{x_k \in \mathcal{S}} exp(-dist(x, x_k)^2)}, \tag{8}$$

where $dist(x, y)$ is the Euclidean distance between the objects x and y.

2.4 Measure of Diversity of Classifiers Ensemble

As it was mentioned previously, the diversity of a classifier ensemble Ψ_E is considered as an independency of the errors made by the member classifiers. Hence the method in which diversity measure is calculated as a variety of class-dependent error probabilities is fully justified.

Similarly, as in competence measure, we assume that at a learning point $x_k \in \mathcal{S}$ the conditional error probability for the class $j \neq \jmath_k$ of the base classifier ψ_l is equal to the appropriate probability of the equivalent RRC, namely:

$$Pe^{(\psi_l)}(j|x_k) = Pe^{(RRC)}(j|x_k). \tag{9}$$

Next, these probabilities can be extended to the entire feature space $\mathcal{X}$ using Gaussian potential function (8):

$$Pe^{(\psi_l)}(j|x) = \frac{\sum_{x_k \in \mathcal{S}, \jmath_k \neq j} Pe^{(\psi_l)}(j|x_k) exp(-dist(x, x_k)^2)}{\sum_{x_k \in \mathcal{S}, \jmath_k \neq j} exp(-dist(x, x_k)^2)}. \tag{10}$$

According to the presented concept, using probabilities (10) first we calculate pairwise diversity at the point $x \in \mathcal{X}$ for all pairs of base classifiers ψ_l and ψ_k from the pool Ψ:

$$D(\psi_l, \psi_k|x) = \frac{1}{M} \sum_{j \in \mathcal{M}} |Pe^{(\psi_l)}(j|x) - Pe^{(\psi_k)}(j|x)|, \tag{11}$$

and finally we get diversity of ensemble of n ($n \leq L$) base classifiers $\Psi_E(n)$ at a point $x \in \mathcal{X}$ as a mean value of pairwise diversities (11) for all pairs of member classifiers, namely:

$$D(\Psi_E(n)|x) = \frac{2}{n \cdot (n-1)} \sum_{\psi_l, \psi_k \in \Psi_E(n); l \neq k} D(\psi_l, \psi_k|x). \tag{12}$$

3 Dynamic Ensemble Selection System

3.1 Method

The proposed DES competence and diversity based classification system (DES-CD) is constructed in the procedure consisting of two steps:

1. For the test object $x \in \mathcal{X}$ and for given ensemble size n and the competence threshold α first the ensemble of classifiers $\Psi_E^*(n)$ is found as a solution of the folowing optimization problem:
$$D(\Psi_E^*(n)|x) = max_{\Psi_E(n)} D(\Psi_E(n)|x) \tag{13}$$
subject to $C(\psi_l|x) \geq \alpha$ for $\psi_l \in \Psi_E^*$. This step eliminates incompetent (inaccurate) classifiers and keeps the ensemble maximally diverse.
2. The selected classifiers are combined by weighted majority voting where the weights are equal to the competence values. The weighted vector of class supports of DES-CD system is given by
$$d_j^{(DES-CD)}(x) = \sum_{\psi_l \in \Psi_E^*(n)} C(\psi_l|x)\, d_{jl}(x) \tag{14}$$
and final decision is made according to the maximum rule (2).

3.2 Solution of Optimization Problem

The key moment in the method developed is the optimization problem (13). As a solution method we propose suboptimal procedure which is followed sequential forward feature selection method [13]. In this method first the set of competent classifiers (better than threshold α) is created and next classifiers are sequentially selected from this set: at first the classifier with the highest competence is chosen, next to the already selected classifier we add another one so as to create the couple with the best diversity, then the three classifiers with the highest diversity, including the selected first and second ones are chosen and so one. This procedure is continued up to n classifiers are selected.

The pseudo-code of the algorithm is as follows:

Input data: $\mathcal{S}$ - learning set; Ψ_L - the pool of classifiers;
n - the size of enesemble; $x \in \mathcal{X}$ - the testing point;
α - the threshold of competence

1. For each $\psi_l \in \Psi_L$ calculate competence $C(\psi_l|x)$ at the point x
2. Create temporal set of competent classifiers at the point x
$\Psi(x) = \{\psi_l \in \Psi_L : C(\psi_l|x) \geq \alpha\}$
3. $\Psi_E^*(n) = \{\psi_{(1)}\}$ and $\Psi(x) = \Psi(x) - \psi_{(1)}$ where
$\psi_{(1)} : C(\psi_{(1)}|x) = max_{\psi \in \Psi(x)} C(\psi|x)$
4. For $i = 2$ to n do
    a) Find $\psi_{(i)} \in \Psi(x)$ for which
    $D(\Psi_E^*(n) \cup \psi_{(i)}|x) = max_{\psi \in \Psi(x)} D(\Psi_E^*(n) \cup \psi|x)$
    b) $\Psi(x) = \Psi(x) - \psi_{(i)}$
    c) $\Psi_E^*(n) = \Psi_E^*(n) \cup \psi_{(i)}$
endfor

4 Experiments

4.1 Databases and Experimental Setup

The benchmark databases used in the experiments were obtained from the UCI Machine Learning Repository (*Breast Cancer Wisconsin*, *Glass*, *Iris*, *Sonar*, *Ionosphere*) and StatLib statistical datasets (*Biomed*). The experiments were conducted in MATLAB using PRTools, which automatically normalizes feature vectors for zero mean and unit standard deviation and for a given $x \in \mathcal{X}$ produces classifying functions for all base classifiers according to the paradigms of their activity [9]. The training and testing datasets were extracted from each database using two-fold cross-validation. The base classifiers and both competence and diversity measures were trained using the same training dataset.

The DES-CD system was compared against three multiclassifier systems: (1) SB system – this system selects the single best classifier in the pool; (2) MV system – this system is based on majority voting of all classifiers in the pool; (3) DES-SC system – this system defines the competence of the classifier ψ for the test object x according to (8) and next the ensemble of competent (better-than-random) classifiers is selected [17] – the final decision is made as in (14).

Two types of classifier ensembles were used in the experiments: homogeneous and heterogeneous. The homogeneous ensemble consisted of 20 pruned decision tree classifiers with Gini splitting criterion. Each classifier was trained using randomly selected 70% of objects from the training dataset.

The pool of heterogeneous base classifiers using in the experiments, consisted of the following nine classifiers [8]: (1-2) linear (quadratic) discriminant classifier based on normal distributions with the same (different) covariance matrix for each class; (3) nearest mean classifier; (4-6) k-NN - k-nearest neighbours classifiers with $k = 1, 5, 15$; (7-8) Parzen classifier with the Gaussian kernel and the optimal smoothing parameter h_{opt} (and the smoothing parameter $h_{opt}/2$); (9) pruned decision tree classifier with Gini splitting criterion.

The following values of parameters of DES-CD system were adopted in the experiments: $\alpha = 1/M$ and $n = max\{\frac{1}{2}|\Psi(x)|, 2\}$.

4.2 Results and Conclusion

Classification accuracies (i.e. the percentage of correctly classified objects) were calculated for the MCSs as average values obtained over 10 runs (5 replications of two-fold cross validation). Statistical differences between the performances of the DES-CD system and the three MCSs were evaluated using Dietterich's 5x2cv test [7]. The level of $p < 0.05$ was considered statistically significant. The results obtained for the MCSs using heterogeneous and homogeneous ensembles are shown in Table 1. For each database and for DES systems the mean sizes of classifier ensembles are given under the classification accuracy.

These results imply the following conclusions:

1. The DES-CD system outperformed the SB and MV classifiers by 7.82% and 5.78 % for heterogeneous ensemble and by 3.99% and 0.47% for homogeneous ensemble, respectively;

Table 1. Classification accuracies of the MCSs using heterogeneous/homogeneous ensembles. The mean sizes of classifier ensembles and statistically significant differences found are given under the classification accuracies. The best result for each database is bolded.

Database	SB (1)	MV (2)	DES-CS (3)	DES-CD (4)
Breast C.W.	95.39/94.99	96.45/95.99	98.03/96.18	**98.05/96.22**
			9.13/19.58	5.21/9.82
			1, 2/1	1, 2/1
Biomed	84.10/83.29	87.62/86.90	**90.09/86.91**	**90.09**/86.81
			8.70/17.31	5.96/9.61
			1, 2/1	1, 2/1
Glass	71.80/61.56	69.55/71.03	**76.46/73.18**	73.26/72.06
			9.46/19.44	5.13/9.75
			1, 2, 4/1, 2	1, 2/1, 2
Iris	96.00/91.07	96.80/90.80	96.67/90.8	**97.33/91.13**
			8.87/20.00	4.98/10.00
				1/
Sonar	74.48/70.19	76.44/76.06	**82.29/77.12**	81.52/**77.12**
			8.63/19.76	5.21/10.00
			1, 2/1, 2	1, 2/1, 2
Ionosphere	84.84/88.15	86.50/89.63	**90.14**/89.74	89.80/**89.88**
			9.06/19.85	5.48/9.95
			1, 2/1	1, 2/1, 2
Average	81.02/81.54	83.06/85.06	**88.94/85.56**	88.84/85.53
			8.97/19.32	5.32/9.85

2. The DES-CD system achieved the highest classification accuracy for 3 datasets and 4 datasets for heterogeneous and homogeneous ensembles, respectively; it produced statistically significant higher scores in 19 out of 36 cases.
3. There are no statistically significant difference between classification accuracies of the DES-CS and the DES-CD systems;
4. The relative difference between mean ensemble sizes for the DES-CS and the DES-CD systems is on average equal to 40.6% and 49% for heterogeneous and homogeneous ensembles, respectively.

5 Conclusion

In this paper a novel procedure for dynamic ensemble selection is proposed using probabilistic measures of competence and diversity of member classifiers. Results of experimental investigations indicate, that the proposed method can eliminate weak classifiers and keep the ensemble maximally diverse. This approach leads to the final classification accuracy which, on average, was very close to the accuracy of DES system using only the competence measure but achieved by means of smaller number of classifiers in the ensemble.

References

1. Aksela, M.: Comparison of classifier selection methods for improving committee performance. In: Windeatt, T., Roli, F. (eds.) MCS 2003. LNCS, vol. 2709, pp. 84–93. Springer, Heidelberg (2003)
2. Aksela, M., Laaksonen, J.: Using diversity of errors for selecting members of a committee classifier. Pattern Recognition 39, 608–623 (2006)
3. Berger, J.O.: Statistical Decision Theory and Bayesian Analysis. Springer, New York (1987)
4. Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: a survey and categorisation. Information Fusion 6, 5–20 (2005)
5. Canuto, A., Abreu, M., et al.: Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensemble. Pattern Recognition Letters 28, 472–486 (2007)
6. Didaci, L., Giacinto, G., Roli, F., Marcialis, G.: A study on the performances of dynamic classifier selection based on local accuracy estimation. Pattern Recognition 38, 2188–2191 (2005)
7. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10, 1895–1923 (1998)
8. Duda, R., Hart, P., Stork, G.: Pattern Classification. John Wiley and Sons, New York (2000)
9. Duin, R., Juszczak, P., Paclik, P., et al.: PRTools4. In: A Matlab Toolbox for Pattern Recognition, Delft University of Technology (2007)
10. Eulanda, M., Santos, D., Sabourin, R., Maupin, P.: A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. Pattern Recognition 41, 2993–3009 (2008)
11. Giacinto, G., Roli, F.: Dynamic classifier selection based on multiple classifier behaviour. Pattern Recognition 34, 1879–1881 (2001)
12. Huenupan, F., Yoma, N., et al.: Confidence based multiple classifier fusion in speaker verification. Pattern Recognition Letters 29, 957–966 (2008)
13. Kanal, L.: Patterns in pattern recognition. IEEE Trans. Information Theory 20, 697–722 (1974)
14. Ko, A., Sabourin, R., Britto, A.: From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition 41, 1718–1733 (2008)
15. Kuncheva, I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley Interscience, Hoboken (2004)
16. Smits, P.: Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection. IEEE Trans. on Geoscience and Remote Sensing 40, 717–725 (2002)
17. Woloszynski, M., Kurzynski, M.: A measure of competence based on randomized reference classifier for dynamic ensemble selection. In: 20th Int. Conf. on Pattern Recognition, pp. 4194–4197. IEEE Computer Press, Istanbul (2010)
18. Woloszynski, T.: Matlab Central File Enchange (2010), http://www.mathwork.com/matlabcentral/fileenchange/28391-classifier-competence-based-on-probabilistic-modeling
19. Woods, K., Kegelmeyer, W., Bowyer, W.: Combination of multiple classifiers using local accuracy estimates. IEEE Trans. on Pattern Analysis and Machine Learning 19, 405–410 (1997)

Complexity and Multithreaded Implementation Analysis of One Class-Classifiers Fuzzy Combiner

Tomasz Wilk and Michał Woźniak

Department of Systems and Computer Networks, Wroclaw University of Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Michal.Wozniak@pwr.wroc.pl

Abstract. More recently, neural network techniques and fuzzy logic inference systems have been receiving an increasing attention. At the same time, methods of establishing decision by a group of classifiers are regarded as a general problem in various application areas of pattern recognition. Similarly to standard two-class pattern recognition methods, one-class classifiers hardly ever fit the data distribution perfectly. The paper presents fuzzy models for one-class classifier combination, compares their computational and expected space complexity with the one from ECOC and decision templates, presents possibility to speed up a fuser processing by means of multithreading.

Keywords: fuzzy logic, ECOC, decision templates, computational complexity, expected space complexity.

1 Introduction

Multiple classifier systems (MCSs) can be considered as a valuable path to achieve more efficient and accurate recognition algorithms and they attain a lot of focus nowadays. The main achievement devoted to MCSs is that they make possible to avoid selection of the worst classifier [1], to obtain the most sensible solution in problems where for some reasons different training data sets exist [2], to help solving instability issue of some classification algorithms as neural networks with random initial weights [2] as well as to use different domains of competence [3]. The binary classification problem can be considered as the basic classification problem. Connecting such classifiers aims to solving multi-class problem by dividing it into dichotomies. In literature there are examples of construction of the multi-class classifier by combining the outputs of two-class classifiers [4, 5]. Usually the combination is made on the basis of a simple nearest-neighbor rule, which finds the class that is closest in some sense to the outputs of the binary classifiers. The most common variations of binary classifier combinations are: one-against-one and one-against-all [5]. Dieterich and Bakiri [6] propose a combination model called ECOC (Error Correcting Output Codes), which in case of binary classifiers ensembles appeared to be a good extension of before mentioned approaches. Passerini *et al.* [7] use successfully this scheme for support vector machines. On the other hand, the combination of one class classifiers still awaits of proper attention [8]. One-class

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 237–244, 2011.

classification problem, also called data description, is a special case of binary classification [9]. Its main goal is to detect anomaly or a state other than the one for the target class [9].

There are several well-established methodologies, design principles, detail algorithms and carefully reported case studies related to hybrid neuro-fuzzy design environments and ensuing architectures [10, 11]. Fuzzy logic (FL) attracts a lot of interest in multiple research domains such as computer science, production support, data classification, diagnostics, data analysis and data mining [10, 12, 13]. One of the most popular data mining tasks is the classifier design. Kuncheva *et al.* [13] prove that decision templates (DT) supported by the fuzzy logic give satisfactory results. Fuzzy systems implemented as adaptive neural networks (ANNs) are the ones that use neural networks support in their properties determination process (for both fuzzy sets and rules). A good example of such system is the Adaptive Neuro-Fuzzy Inference System (ANFIS), which provided good results in modeling of non-linear functions. ANFIS learns the membership function parameters from a data set with features related to a given problem [12].

The purpose of our contribution is to present a theoretical model of fuzzy combination method of one-class classifiers. Support values from corresponding elementary classifiers are fuzzified and in the next step combined using Sugeno rules. Final supports are achieved through the fuzzy inference system and the class label with the highest value of support is assigned to the input object. Fuzzy equivalent of both ECOC and DT is also investigated and presented. Computational complexity and expected space complexity are calculated and multithreaded implementation model is discussed. The promising tests results of the proposed fuzzy combiner model can be found in [14].

2 Proposed Fuzzy Combination Methods

2.1 Fuzzy Combiner Model (FC)

Let us denote $M = \{1,2,...,M\}$ as both the set of class labels in the problem at hand, and the quantity of one-class classifiers. $g_i(x)$ comprises the support value for the statement that the current input data belongs to the *i*-th class. Let us define the set of elementary classifiers as $\Psi = \{\Psi_1,..,\Psi_i,..,\Psi_M\}$, where Ψ_i means the *i*-th classifier in the ensemble. Furthermore, we can describe our fuzzy combiner as a function $l(\Psi(x)) = Y$, where $Y = \{y_1,...,y_M\}$, and $y_j \in R^n$. Therefore, y_i corresponds to the final support for the *i*-th class given by the fuzzy combiner. According to the proposed approach, the label of each input object is determined from the aggregation of the classifiers in form of one-class classifiers. In the proposed model both one-class classifiers and 'one-against-all' binary classifiers will be used.

The elementary classifiers are assumed to be trained and this issue will not be investigated further. Support function values returned by classifiers are interpreted using fuzzy logic. It is a common practice that a Fuzzy Decision Making System (FDMS) usually comprises four main components: the fuzzification block, the reasoning block, the knowledge base and finally the defuzzification block.

Explanation of the given blocks can be found in [11]. According to Rutkowski, the fuzzy set A in a certain non-empty space can be described as:

$$A_i(v) = \left\{\left(v_i, \mu_{A_i}(v_i)\right): \mu_{A_i}(v_i) \in [0,1]\right\} \quad \text{where } v_i = g_i(x) \tag{1}$$

and μ_A represents the membership function in relation to the fuzzy set A. The main idea of the equation is to describe the fuzzified discriminant function. Männle [15] points out that the shape of the membership function has a minor influence on the FIS performance. For computational reasons triangular shape was chosen.

Theoretically, a higher number of fuzzy sets should improve quality, but it would be at the cost of the performance of proposed model. It could also decrease generalization properties of the fuzzy combiner and provide additional parameters of the model, for which larger training dataset would be required. One can also consider fuzzy rules pruning, as described in [16, 17]. We propose the use of fuzzy subtractive clustering [18] in order to find membership function parameters and fuzzy rules candidates. As a result the number of the fuzzy rules will not be a parameter of the proposed system.

It has already been proved [15] that the Sugeno-type fuzzy rules provide lower magnitude of errors in comparison to those of Yasukawa. Thus, in the present case the Sugeno-derived rules will be applied as follows:

$$\begin{gathered}\text{IF}(\, g_1(x) \text{ IS } A_1^k \text{ AND/OR } \ldots \text{ AND/OR } g_M(x) \text{ IS } A_M^k \,) \text{ THEN} \\ y_i^{(k)}(\Psi(x)) = F_k(\Psi(x)) = c_0^{(k)} + c_1^{(k)} g_1(x) + \ldots + c_M^{(k)} g_M(x)\end{gathered} \tag{2}$$

where k expresses the number of the fuzzy rule and each $c_0^{(k)},\ldots,c_M^{(k)} \in R$.

Each of its outputs is distinctive with the FL engine of the Sugeno-type and can be written as:

$$y_i(\Psi(x)) = \frac{\sum_{k=1}^{S} (W_k(\Psi(x)) * F_k(\Psi(x)))}{\sum_{k=1}^{S} W_k(\Psi(x))} \tag{3}$$

where S corresponds to the rules number, W_k - fuzzy relation on the k-th rule, and F_k - response of the k-th linear function. The product operation was chosen as the Cartesian product and t-norm, because of its differentiability [11]. The proposed fuzzy combiner selects the class for which the support value from the equation (3) reaches the highest magnitude. Both FL features and rule sets are evaluated for a given classification problem with support of ANNs [10]. We use ANFIS (*Artificial Neuro-Fuzzy Inference System*) belonged to the class of adaptive networks. The detailed model description can be found in [14].

2.2 Fuzzy ECOC and Fuzzy DT

Both models, ECOC and DT, are based on some sort of a distance measure from a template. In the case of DT with a symmetric distance an assumption is made that

supports for given classes are already fuzzified. Such plank gives good results, but does not have to be true. Weighting of the classifier responses for dichotomies is another field for improvement. We will refer to DTs as both decision templates with Euclidean and symmetric distance. We propose to improve DTs and ECOC for the binary classifiers by use of fuzzy logic to process the template. The proposed models will be called FDT and FECOC respectively. Both models will use Gaussian function as a membership function. Additionally, ANN is not enabled in the proposed models and so there are no computational limitations, therefore FL engine of Mamdani-type can be applied.

Its advantades incorporate full fuzzy logic inference as also operation on multiple outputs. Rules parameters are assigned basing on the template, each output signal has two fuzzy sets assigned, described by linguistic variables LOW and HIGH, correspondingly. Both Cartesian product and the used t-norm are a product operation. Example of the proposed model is presented below. Let us assume a problem with M=3 and implementation of FECOC. One can easily recognize the ECOC matrix in it, expanded by the column with class numbers, for which the given template was generated. Output signals in the process of fuzzification are transformed into two fuzzy sets, both with triangular membership functions, with apex at position 0 (LOW) and 1 (HIGH), and both reach 0 value at opposite tops (1 and 0 input value, correspondingly). Our template is shown in Fig. 1.

1	0	0	1
0	1	0	2
0	0	1	3

Fig. 1. Template for fuzzy ECOC where M=3

For the first row of the presented template the fuzzy rule is of form:

$$\begin{aligned} &\text{IF } (\ g_1(x) \text{ IS } A_1^1 \text{ AND } \ldots \text{ AND } g_3(x) \text{ IS } A_3^1\ ) \text{ THEN} \\ &\quad y_1 \text{ IS } D_1^1 \text{ AND } y_2 \text{ IS } D_2^2 \text{ AND } y_3 \text{ IS } D_2^3 \end{aligned} \tag{4}$$

where D_1 corresponds to linguistic variable HIGH and D_2 to LOW. Identical reasoning is applied to the remaining rules. The detailed description of FECOC and FDT can be found in [19].

3 Computational Complexity and Expected Space Complexity

In case of the FC algorithm, the expected space complexity can be represented as the following:

$$O_{enp} = M * (2 * 8 * Q + S * M * 1 + 2 * 8 * M) == M * (16 * Q + M * (S + 16)) \tag{5}$$

where Q defines a number of fuzzy sets per input. For simplicity, an assumption is made that for each input in every of the Sugeno type inference system, this number remains constant. Similar assumption is made in relation to the number of fuzzy rules

in the applied fuzzy engines. Also it is assumed that 8 bytes are needed to store values and 1 byte numbers are required to store indexes of the used fuzzy sets/membership functions in the equation.

The expected space complexity of ECOC equals $O_{enp} = M^2 * 1$ and that of DTs - $O_{enp} = M^3 * 8$.

In both FECOC, and FDT we do not have to store parameters of the output fuzzy sets and membership functions and M=S=Q, thus, the expected space complexity can be written as:

$$O_{enp} = 2 * 8 * Q + S * M * 1 = 16 * Q + M * S = 16 * M + M^2 \tag{6}$$

The amount of memory needed to store code matrixes (decision templates) for ECOC, and DTs is relatively lower than in FC; FECOC and FDT have comparable expected space complexity. Taking into consideration the current state of development of mass memories the before mentioned models can be used in most of the real pattern recognition problems.

Computational complexity for the FC combiner can be expressed as the following:

$$O(M,S) = O(\arg\max_{i \in M}) * O\left(\sum_{s=1}^{S}(W_s(\Psi(x)) * F_s(\Psi(x)))\right) * O\left(\sum_{k=1}^{S} W_s(\Psi(x))\right) \tag{7}$$

One must remember that $W_s = \prod_{m=1}^{M} \mu_m^s$. As we use membership function of triangular shape, its computational complexity can be considered as a constant value and its significance for the total complexity estimation is negligible. Fuzzy relation of the product form has a complexity of $O(n)$, the same as the one for the summation of weights. Thus, we achieve

$$O\left(\sum_{s=1}^{S} W_s(\Psi(x))\right) = O(S) * O(M),\ O(F_s(\Psi(x))) = O(M) + O(M) = O(M),$$

and $O\left(\sum_{s=1}^{S}(W_s(\Psi(x)) * F_s(\Psi(x)))\right) = O(S) * O(M^2)$.

Finally, the computational complexity of the FC is

$$O(M,S) = O(M) * O(S) * O(M) * O(M^2) * O(S) = O(M^4) * O(S^2) \tag{8}$$

Such a formula was achieved by assuming that the number of rules in each fuzzy engine is equal, which is not a necessary condition.

In both ECOC and DTs, M binary classifiers are used. Their computational complexity can be expressed as:

$$O_{ECOC}(M) = O_{DT}(M) = O(M^2) \tag{9}$$

As for fuzzy extensions, namely, FECOC and FDT, we get $O(M,S) = O(\arg\max_{i \in M}) * (O(y_i(\Psi)) * O(y_i^k(\Psi)))$ and specifically $O(y_i^k(\Psi)) = O(M)$

as we use a product operation as a t-norm. Finally, computational complexity for FECOC and FDT is of a form

$$O(M,S) = O(M) * (O(M) * O(M)) = O(M^3) \tag{10}$$

All three proposed models have higher computational complexity than ECOC and DTs. By means of multiprocessor machine we could decrease the complexity of FC model by multithreading. We will assume that the threads quantity at our disposal is equal to *M*.

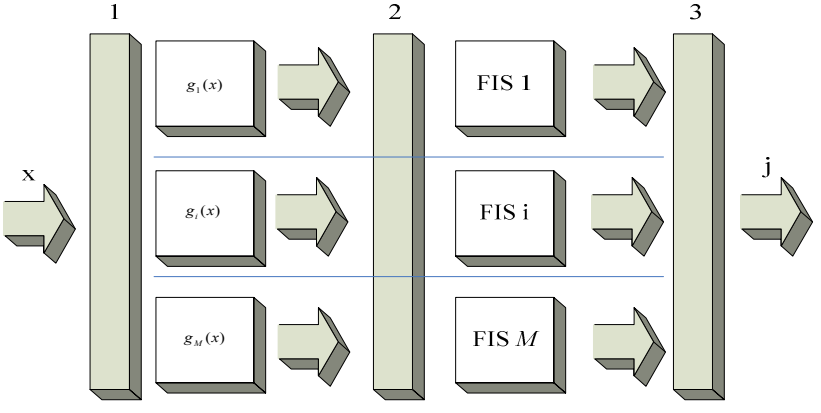


Fig. 2. Multithreaded implementation of the FC algorithm

The multithreaded implementation of the FC algorithm (Fig. 2.) has three blocking points, in which further processing is not possible until similar processing is finished in all other threads. In the second blocking point we still can fuzzify supports form the classifiers, in which processing ended earlier. However to proceed with the fuzzy reasoning we need all input signals to be fuzzified. In the FC model, after each fuzzy inference engine generates response, we select the highest support value and assign the input object to the class label with its highest value. Therefore, it is possible to improve computational complexity to a complexity not higher than of a single fuzzy inference engine.

In case of both DT and ECOC (Fig. 3.) possible improvement of computational complexity is more significant in multithreaded scenario. We are able to process most of combiner operations the moment we achieve the *i*-th support value from the *i*-th classifier in the ensemble. In ECOC, we are able to process the *i*-th column of the code matrix, that is $-|g_i(x) - C(i)|$ where $C(i)$ is the *i*-th column of the code matrix. Next results for each column are to be added to achieve final supports for the classes. It can be done in the first thread which finishes processing. The same improvement can be expected from considered DTs. Computational complexity can be reduced to $O_{ECOC}(M) = O_{DT}(M) = O(M) + O(M) = O(M)$.

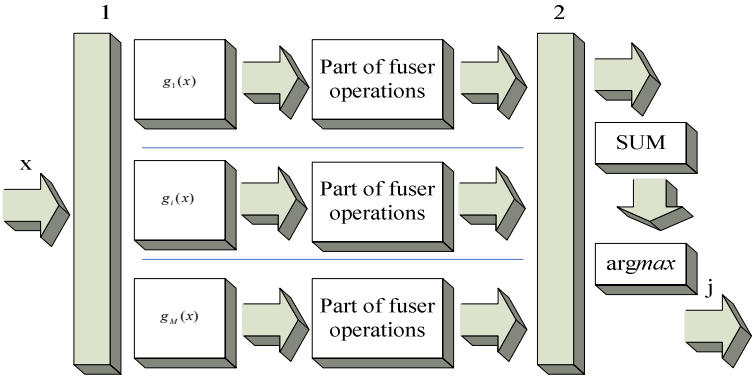


Fig. 3. Multithreaded implementation of DT/ECOC algorithms

4 Conclusions

The fuzzy combiner methods for one class classifiers, their expected space complexities and computational complexities were presented. All described models are computationally expensive but it can be reduced by means of multithreaded implementation. Especially it may give good results in case of ECOC and DTs. As next step authors consider implementing such extension to validate the concept.

Acknowledgments

This work is supported by The Polish Ministry of Science and Higher Education under the grant which is realizing in years 2010-2013.

References

1. Marcialis, G.L., Roli, F.: Fusion of Face Recognition Algorithms for Video-Based Surveillance Systems. In: Multisensor Surveillance Systems: The Fusion Perspective. Kluwer Academic Publishers, Dordrecht (2003)
2. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence 1(22), 4–37 (2000)
3. Wolpert, D.H.: The supervised learning no-free lunch Thorem. In: Proc. Of the 6th On-line World Conference on Soft-Computing in Industrial Application (2001)
4. Tax, D.M.J., Duin, R.P.W.: Using two-class classifiers for multiclass classification. Pattern Recognition Group 2, 124–127 (2002)
5. Duan, K., Keerthi, S.S., Chu, W.: Multi-Category Classification by Soft-Max Combination of Binary Classifiers. In: 4th International Workshop, June 11-13, pp. 125–134 (2003)
6. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263–286 (1995)
7. Passerini, A., Pontil, M., Frasconi, P.: New Results on Error Correcting Output Codes of Kernel Machines. Neural Networks 15, 45–54 (2004)

8. Giacinto, G., Perdisci, R., Del Rio, M., Roli, F.: Intrusion detection in computer networks by a modular ensemble of one-class classifiers. Information Fusion 9, 69–82 (2009)
9. Tax, D.M.J., Duin, R.P.W.: Combining One-class Classifiers. In: Proceedings of Multiple Classifier Systems, pp. 299–308 (2001)
10. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, New Jersey (2004)
11. Rutkowski, L.: Flexible Neuro-Fuzzy Systems. Neural Networks 14(3), 554–574 (2003)
12. Güler, I., Übeyli, E.D.: Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients. Journal of Neuroscience Methods 2(148), 113–121 (2005)
13. Kuncheva, L.I., Bezdek, J.C.W., Duin, R.P.: Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition 2, 299–314 (2001)
14. Wilk, T., Woźniak, M.: Combination of one-class classifiers for multiclass problems by fuzzy logic. Neural Network World International Journal on Non-Standard Computing and Artificial Intelligence 20, 853–869 (2010)
15. Männle, M.: Parameter Optimization for Takagi-Sugeno Fuzzy Models - Lessons Learnt. In: Proceedings of IEEE Systems Man and Cybernetics, pp. 111–116 (2001)
16. Pal, N.R., Pal, T.: On rule pruning using fuzzy neural networks. Fuzzy Sets and Systems 106(3), 335–347 (1999)
17. Kim, D.-W., Park, J.-B., Joo, Y.-H.: Design of Fuzzy Rule-Based Classifier: Pruning and Learning. In: Wang, L., Jin, Y. (eds.) FSKD 2005. LNCS (LNAI), vol. 3613, pp. 416–425. Springer, Heidelberg (2005)
18. Mahmoud, T.S., Marhaban, M.H., Hong, T.S.: ANFIS Controller with Fuzzy Subtractive Clustering Method to Reduce Coupling Effects in Twin Rotor MIMO System (TRMS) with Less Memory and Time Usage. In: Advanced Computer Control, pp. 19–23 (2009)
19. Wilk, T., Woźniak, M.: Soft computing methods applied to combination of one-class classifiers. Neurocomputing (in press)

Costs-Sensitive Classification in Multistage Classifier with Fuzzy Observations of Object Features

Robert Burduk

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

Abstract. In the paper the problem of cost in hierarchical classifier is presented. Assuming that both the tree structure and the feature used at each non-terminal node have been specified, we present the expected total cost for two cases. The first one concerns the non fuzzy observation of object features, the second concerns the fuzzy observation. At the end of the work the difference between expected total cost of fuzzy and non fuzzy data is determined. Obtained results relate to the locally optimal strategy of Bayes multistage classifier.

1 Introduction

The problem of cost-sensitive classification is broadly discussed in literature. The major costs are the costs of feature measurements (tests costs) and the costs of classification errors. Cost of feature measurements are discussed in [1], [2], [3], [4] other works included the cost of classification errors [5], [6]. In a more realistic setup, there are good reasons for considering both the costs of feature measurements and the costs of classification errors. For example, there should be a balance between cost of measuring each feature and the contribution of the test to accurate classification. It often happens that the benefits of further classification are not worth the costs of feature measurements. This means that a cost must be assigned to both the tests and the classification errors. In the works [7], [8], [9] both feature costs and misclassification cost are taken into account.

In this paper, we consider the costs in hierarchical classifier. In our model of pattern recognition we use the Bayes rule in each internal node of decision tree [10]. We consider the problem of classification for the case in which the observations of the features are represented by the non fuzzy or fuzzy sets [11]. Additionally, we consider the locally optimal strategy of multistage recognition task where the zero-one loss function is taken into account. The obtained costs of non fuzzy observations are compared with the case where observations are fuzzy. The difference in costs of those two cases is the subject of this paper.

The contents of the work are as follows: Section 2 introduces the necessary background and describes the Bayes hierarchical classifier. In section 3, the recognition algorithm with fuzzy observations is presented. In section 4, we present

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 245–252, 2011.

the cost model and the difference between the costs of the non fuzzy and fuzzy observations of features in Bayes hierarchical classifier.

2 Bayes Hierarchical Classifier

In the paper [12], the Bayesian hierarchical classifier is presented. The synthesis of a multistage classifier is a complex problem. It involves specification of following components [13]:

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

This paper is devoted only to the last problem. This means that we will only consider the presentation of decision algorithms, assuming that both the tree structure and the feature used at each non-terminal node have been specified.

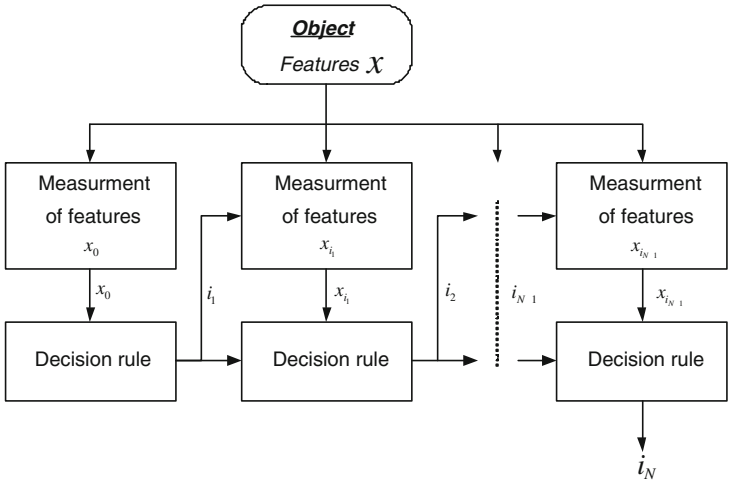


Fig. 1. Block diagram of the hierarchical classifier

The procedure in the Bayesian hierarchical classifier consists of the following sequences of operations presented in Fig. 1. At the first stage, some specific features x_0 are measured. At this point we assume the feature acquisition costs associated with this stage. They are chosen from among all accessible features x, which describe the pattern that will be classified. These data constitute a basis for making a decision i_1. This decision, being the result of recognition at the first stage, defines a certain subset in the set of all classes and simultaneously indicates features x_{i_1} (from among x) which should be measured in order to make a decision at the next stage.

Now, at the second stage, features x_{i_1} are measured, which together with i_1 constitutes a basis for making the next decision i_2. This decision, – like i_1 –

indicates features x_{i_2} that are necessary to make the next decision (at the third stage, as in the previous stage) which define in turn a certain subset of classes, not in the set of all classes, but in the subset indicated by the decision i_2, and so on. The whole procedure ends at the N-th stage, where the decision made i_N indicates a single class, which is the final result of multistage recognition.

2.1 Decision Problem Statement

Let us consider a pattern recognition problem, in which the number of classes equals M. Let us assume that the classes are organized in a $(N+1)$ horizontal decision tree. Let us number all the nodes of the decision-tree constructed with consecutive numbers of $0, 1, 2, \ldots$, reserving 0 for the root-node, and let us assign numbers of classes from the $\mathcal{M} = \{1, 2, \ldots, M\}$ set to terminal nodes so that each of them can be labelled with the class number connected with that node. This allows us to introduce the following notation:

- $\mathcal{M}(n)$ – the set of nodes, whose distance from the root is n, $n = 0, 1, 2, \ldots, N$. In particular $\mathcal{M}(0) = \{0\}$, $\mathcal{M}(N) = \mathcal{M}$,
- $\overline{\mathcal{M}} = \bigcup_{n=0}^{N-1} \mathcal{M}(n)$ – the set of internal nodes (non terminal),
- $\mathcal{M}_i \subseteq \mathcal{M}(N)$ – the set of class labels attainable from the i-th node ($i \in \overline{\mathcal{M}}$),
- $\mathcal{M}^i$ – the set of nodes of immediate descendant node i ($i \in \overline{\mathcal{M}}$),
- m_i – node of direct predecessor of the i-th node ($i \neq 0$),
- $s(i)$ – the set of nodes on the path from the root-node to the i-th node, $i \neq 0$.

We will continue to adopt the probabilistic model of the recognition problem, i.e. we will assume that the class label of the pattern being recognized $j_N \in \mathcal{M}(N)$ and its observed features x are realizations of a couple of random variables $\boldsymbol{J}_N$ and $\boldsymbol{X}$. The complete probabilistic information denotes the knowledge of a priori probabilities of classes [14]:

$$p(j_N) = P(J_N = j_N), \quad j_N \in \mathcal{M}(N) \tag{1}$$

and class-conditional probability density functions:

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

Let

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

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

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

$$\Psi_i : X_i \to \mathcal{M}^i, \quad i \in \overline{\mathcal{M}}\ . \tag{4}$$

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

$$D^k_{x_i} = \{x_i \in X_i : \Psi_i(x_i) = k\},\ k \in \mathcal{M}^i, \quad i \in \overline{\mathcal{M}}. \tag{5}$$

Our aim is to minimize the expected risk function (expected loss function $L(I_N, J_N)$) denoted by:

$$R^*(\pi_N) = \min_{\pi_N} R(\pi_N) = \min_{\pi_N} E[L(I_N, J_N)]. \tag{6}$$

where π_N is the strategy of the decision tree classifier. The π_N is a set of classifying rules used at a particular node $\pi_N = \{\Psi_i\}_{i \in \overline{\mathcal{M}}}$.

We consider the locally optimal strategy $\bar{\pi}_N$. This strategy is derived in order to minimize the local criteria , which denote probabilities of misclassification of particular nodes of a tree. Its decision rules are mutually independent. The recognition algorithm at the n-th stage is as follows:

$$\bar{\Psi}_{i_n}(x_{i_n}) = i_{n+1} \quad \text{if} \quad i_{n+1} = \arg \max_{k \in \mathcal{M}^{i_n}} p(k) f_k(x_{i_n}). \tag{7}$$

3 The Recognition Algorithm with Fuzzy Observations

Fuzzy number A is a fuzzy set defined on the set of real numbers $\mathbb{R}$ characterized by means of a membership function $\mu_A(x)$, $\mu_A : \mathbb{R} \to [0, 1]$. In this study, the special kinds of fuzzy numbers including triangular fuzzy numbers are employed. Triangular fuzzy numbers can be defined by a triplet $A = (a_1, a_2, a_3)$.

Fuzzy information $\mathcal{A}_k \in \Re^d$, $k = 1, ..., d$ (d is the dimension of the feature vector) is a set of fuzzy events $\mathcal{A}_k = \{A^1_k, A^2_k, ..., A^{n_k}_k\}$ characterized by membership functions

$$\mathcal{A}_k = \{\mu_{A^1_k}(x_k), \mu_{A^2_k}(x_k), ..., \mu_{A^{n_k}_k}(x_k)\}. \tag{8}$$

The value of index n_k defines the possible number of fuzzy events for x_k (for the k-th dimension of feature vector). In addition, let us assume that for each observation subspace x_k the set of all available fuzzy observations (8) satisfies the orthogonality constraint [15]:

$$\sum_{l=1}^{n_k} \mu_{A^l_k}(x_k) = 1. \tag{9}$$

The probability of fuzzy event is assumed in Zadeh's form [16]:

$$P(A) = \int_{\Re^d} \mu_A(x) f(x) dx. \tag{10}$$

The probability $P(A)$ of a fuzzy event A defined by (10) represents a crisp number in the interval $[0, 1]$.

Now applying a procedure similar to [12] and using the zero-one loss function, we obtain the searched locally optimal strategy with decision algorithms as follows [10]:

$$\overline{\Psi}_{i_n}(A_{i_n}) = i_{n+1} \quad \text{if} \tag{11}$$

$$i_{n+1} = \arg \max_{k \in \mathcal{M}^{i_n}} p(k) \int_{\Re^{i_n}} \mu_{A_{i_n}}(x_{i_n}) f_k(x_{i_n}) dx_{i_n}.$$

4 Cost Model

In the costs model we specify two costs. There are the feature acquisition costs and the misclassification costs. We assume that the feature acquisition cost for each internal node is known for non fuzzy observations. It means that i-th node has an associated feature acquisition costs $FAC(i)$. In this case, each feature has an independent cost and the cost of a set of features is just an additive cost.

Each patch $s(k), k \in \mathcal{M}$ represents a sequence of ordered feature values and the final classification. Each path has an associated feature acquisition cost. It can be computed as follows:

$$FAC(s(k)) = \sum_{i \in s(k),} \quad \sum_{i \notin \mathcal{M}} FAC(i). \tag{12}$$

Now we present the expected misclassification cost. The purpose of statistical pattern recognition is to minimize the expected risk function (6). In this work we assume that the misclassification costs is represented by the zero-one loss function. This misclassification costs model assigns cost equal 0 for correct classification and equal 1 for incorrect classification. For such a loss function the expected risk function means the probability of error. This probability is interpreted as the expected misclassification cost. The probability of error $Pe(i)$ in node i is given by [12]:

$$Pe(i) = 1 - \sum_{k \in \mathcal{M}^{i_n}} \frac{p(k)}{p(i)} q(k/i, k). \tag{13}$$

Each path has an associated misclassification cost. It can be computed as follows:

$$MC(s(k)) = \sum_{i \in s(k),} \quad \sum_{i \notin \mathcal{M}} Pe(i). \tag{14}$$

The total costs of a path is the sum of the feature acquisition costs and the expected misclassification costs:

$$TC(s(k)) = FAC(s(k)) + MC(s(k)). \tag{15}$$

The expected total cost of the locally optimal strategy $\bar{\pi}_N$ is then the sum of total cost of each path, weighted by the probability of the following path:

$$ETC(\bar{\pi}_N) = \sum_{k=1}^{\mathcal{M}} P(s(k))TC(s(k)). \tag{16}$$

4.1 Cost Model for Fuzzy Observations

For the fuzzy observation we know that the feature acquisition costs increase. This increase is determined by the parameter λ as follows: $FAC_F(i) = (1 + \lambda)FAC(i)$, where $FAC_F(i)$ is the feature acquisition costs for fuzzy observations. For the parameter λ=0 the feature acquisition costs for fuzzy and non fuzzy observations are equal. When $\lambda > 0$ then the feature acquisition costs for fuzzy observations are greater than for non fuzzy observations.

The probability of error $Pe_F(i)$ in node i for fuzzy data is the following [10]:

$$Pe_F(i) = 1 - \sum_{k \in \mathcal{M}^{in}} \frac{p(k)}{p(i)} \sum_{A_i \in D_{x_i}^{(k)}} \int_{\Re^i} \mu_{A_i}(x_i) f_k(x_i) dx_i \tag{17}$$

For such a specific feature acquisition costs and misclassification costs for fuzzy observations we can present the expected total cost of the locally optimal strategy for fuzzy observations:

$$ETC_F(\bar{\pi}_N) = \sum_{k=1}^{\mathcal{M}} P(s(k))TC_F(s(k)), \tag{18}$$

where $TC_F(s(k))$ is calculated by taking into account $FAC_F(i)$ and (17).

When we use fuzzy information on object features instead of exact information we increase the expected total cost. The difference between the expected total cost for the fuzzy $ETC_F(\bar{\pi}_N)$ and non fuzzy data $ETC(\bar{\pi}_N)$ is the following:

$$ETC_F(\bar{\pi}_N) - ETC(\bar{\pi}_N) = \tag{19}$$

$$= \sum_{k=1}^{\mathcal{M}} P(s(k))(TC_F(s(k)) - TC(s(k))) =$$

$$= \sum_{k=1}^{\mathcal{M}} P(s(k))((1+\lambda)FAC(s(k)) + MC_F(s(k)) - (FAC(s(k)) + MC(s(k))))$$

$$= \lambda \sum_{k=1}^{\mathcal{M}} P(s(k))(FAC(s(k)) + \sum_{i \in s(k),\ i \notin \mathcal{M}} (Pe_F(k) - Pe(k))).$$

The expression $Pe_F(k) - Pe(k)$ presents the difference between the probability of misclassification for the fuzzy and non fuzzy data in decision tree node i. In the paper [10], the value of $Pe_F(k) - Pe(k)$ for a full probabilistic information is presented.

5 Conclusion

In the present paper, we have concentrated on the costs of hierarchical classifier. In this study we assume that the decision tree is known, that is, the work does not generate its structure. The study considered two types of costs, the feature acquisition costs and the misclassification costs. An expected total cost of the locally optimal strategy was presented for the assumptions such as zero-one loss function. This cost concerns the case of non fuzzy observations of an object feature. Moreover, the expected total cost of the locally optimal strategy for fuzzy observations of object feature was presented. Towards conclusion of the work the difference between the expected total cost for the fuzzy and non fuzzy data was determined.

In future work we can consider another loss functions. For the Bayes multistage classifier it may be a stage-dependent or node-dependent loss function.

Acknowledgements. This work is supported by The Polish Ministry of Science and Higher Education under the grant which is being realized in years 2010-2013.

References

1. Núñez, M.: The use of background knowledge in decision tree induction. Machine Learning 6(3), 231–250 (1991)
2. Penar, W., Woźniak, M.: Experiments on classifiers obtained via decision tree induction methods with different attribute acquisition cost limit. Advances in Soft Computing 45, 371–377 (2007)
3. Penar, W., Woźniak, M.: Cost-sensitive methods of constructing hierarchical classifiers. Expert Systems 27(3), 146–155 (2010)
4. Tan, M.: Cost-sensitive learning of classification knowledge and its applications in robotics. Machine Learning 13, 7–33 (1993)
5. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and regression trees, California, Wadsworth (1984)
6. Knoll, U., Nakhaeizadeh, G., Tausend, B.: Cost-sensitive pruning of decision trees. In: Proceedings of the Eight European Conference on Machine Learning ECML, vol. 94, pp. 383–386 (1994)
7. Yang, Q., Ling, C., Chai, X., Pan, R.: Test-cost sensitive classification on data with missing values. IEEE Transactions on Knowledge and Data Engineering 18(5), 626–638 (2006)
8. Saar-Tsechansky, M., Melville, P., Provost, F.: Active feature-value acquisition. Management Science 55(4), 664–684 (2009)
9. Turney, P.: Cost-sensitive classificcation: Empirical evaluation of a hybrid genetic decision tree induction algorithm. Journal of Artificial Intelligence Research 2, 369–409 (1995)
10. Burduk, R.: Randomness and fuzziness in Bayes multistage classifier. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS (LNAI), vol. 6076, pp. 532–539. Springer, Heidelberg (2010)
11. Burduk, R.: Classification error in Bayes multistage recognition task with fuzzy observations. Pattern Analysis and Applications 13(1), 85–91 (2010)

12. Kurzyński, M.: On the multistage Bayes classifier. Pattern Recognition 21, 355–365 (1988)
13. Kulkarni, A.: On the mean accuracy of hierarchical classifiers. IEEE Transactions on Computers 27, 771–776 (1978)
14. Kuncheva, L.I.: Combining pattern classifier: Methods and Algorithms. John Wiley, New York (2004)
15. Okuda, T., Tanaka, H., Asai, K.: A formulation of fuzzy decision problems with fuzzy information using probability measures of fuzzy events. Information and Control 38, 135–147 (1978)
16. Zadeh, L.A.: Probability measures of fuzzy events. Journal of Mathematical Analysis and Applications 23, 421–427 (1968)

Fusion of Similarity Measures for Time Series Classification

Krisztian Buza, Alexandros Nanopoulos, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim, Germany
{buza,nanopoulos,schmidt-thieme}@ismll.de

Abstract. Time series classification, due to its applications in various domains, is one of the most important data-driven decision tasks of artificial intelligence. Recent results show that the simple nearest neighbor method with an appropriate distance measure performs surprisingly well, outperforming many state-of-the art methods. This suggests that the choice of distance measure is crucial for time series classification. In this paper we shortly review the most important distance measures of the literature, and, as major contribution, we propose a framework that allows fusion of these different similarity measures in a principled way. Within this framework, we develop a hybrid similarity measure. We evaluate it in context of time series classification on a large, publicly available collection of 35 real-world datasets and we show that our method achieves significant improvements in terms of classification accuracy.

Keywords: time series, classification, fusion, hybrid similarity measure.

1 Introduction

One of the most prominent research topics in artificial intelligence, in particular in data-driven decision tasks, is time series classification. Given a series of measured values, like the blood pressure of a patient every hour, the position coordinates of a ballpoint pen in consecutive moments, acoustic or electrocardiograph signals, etc., the task is to recognize which pre-defined group the signal belongs to. In the previous applications these groups could correspond, for example, to words written or said by a person or to the health status of a patient (normal, high or low blood pressure; regular or irregular heart rhythm). In general, besides speech recognition [15], time series classification finds applications in various domains such as finance, medicine, biometrics, chemistry, astronomy, robotics, networking and industry [8].

Because of the increasing interest in time-series classification, various approaches have been introduced ranging from neural and Bayesian networks to genetic algorithms, support vector machines and frequent pattern mining [2]. One of the most surprising recent results is, however, that the simple 1-nearest neighbor (1-NN) classifier using dynamic time warping (DTW) distance [15] has been shown to be competitive or superior to many state-of-the art time-series

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 253–261, 2011.

classification methods [6], [9], [13]. These results inspired intensive research of DTW in the last decade: this method has been examined in depth (for a thorough summary of results see [11]), while the improvements in its accuracy [2], [12] and efficiency [10] allowed to apply it to large, real-word recognition problems.

This success of DTW suggests that, in time series classification, what really matters is the distance measure, i.e. when and why two time series are considered to be similar. DTW allows shifting and elongations in time series, i.e., when comparing two time series t_1 and t_2, the i-th position of t_1 is not necessarily matched to the i-th position in t_2, but it can be matched to some other positions too (that are usually close to the i-th position). By allowing for shifting and elongations, DTW captures the global similarity of the shape of two time series very well. In general, however, many other characteristic properties might be crucial in a particular application, such as similar global or local behavior in the frequency domain, that can be captured by the Fourier or Cosine-spectrum or the Wavelet Transform of the signal [3], [7].

In this paper, we examine this phenomenon in more detail. We consider a set of state-of-the art time series similarity measures and discuss what kind of similarity they capture. As major contribution, we propose a framework that allows fusion of these different similarity measures in a principled way. Within this framework, we develop a hybrid similarity measure. We evaluate our findings in context of time series classification on a large, publicly available collection of 35 real-world datasets and show that our method achieves substantial (statistically significant) improvements in terms of classification accuracy.

2 Related Work

We focus on related works that are most relevant w.r.t. the major contribution, i.e. fusion of similarity measures. For a (short) review of time series similarity measures we refer to Section 3.

There were many attempts to fuse several classifiers by combining their outputs. This resulting structure is often called as an ensemble of classifiers. Besides the simple schemes of majority and weighted voting, more sophisticated methods were introduced such as bagging, boosting [1], [5] and stacking [18]. Ensembles of classifiers have been designed and applied for time series classification in e.g. [16], [19]. In contrast to these works, we aim at fusing the *similarity measure*, instead of working at the level of classifiers' outputs.

One of the core components of our framework is a model for pairwise decisions about whether two time series belong to the same class. Similar models were applied in context of web page clustering [14] and de-duplication [4]. Both of these works, however, aimed at finding equivalent items (whereas the concept of "equivalence" is understood in a broad sense by defining e.g. two web pages as equivalent if they write about the same person). In contrast to them, we work with time series, and, more importantly, we focus on classification.

Fusion of similarity measures is also related to multiple kernel learning [17]. As opposed to [17], we consider time series in a simple and generic framework.

3 Time Series Similarity Measures

In this section we review the most important time series similarity measures. Please note, that throughout this paper we use *similarity measure* and *distance measure* as synonyms. Denoting the i-th position of the time series t by $t(i)$ we can define the *Euclidean Distance* of two time series t_1 and t_2 of length k as: $d_E(t_1, t_2) = \sqrt{\sum_{i=1}^{k}(t_1(i) - t_2(i))^2}$.

The intuition behind *Dynamic Time Warping* (DTW) is that we can not expect an event to happen (or a characteristic pattern to appear respectively) at *exactly* the same time position and its duration can also (slightly) vary. DTW captures global similarity of two time series' shapes in a way that it allows for shifting and elongations. DTW is an edit distance: the distance of two time series t_1 and t_2 of length k, denoted as $d_{DTW}(t_1, t_2)$, is the cost of transforming t_1 into t_2. This can be calculated by filling entries of a $k \times k$ matrix. See also [11], [12].

The *Discrete Fourier Transformation* (DFT) maps the time series t to a set of (complex) coefficients $\{c_f\}_{f=1}^{k}$ that are defined as

$$c_f(t) = \frac{1}{\sqrt{k}} \sum_{i=1}^{k} t(i) e^{-\frac{2\pi j f i}{k}}.$$

where $j = \sqrt{-1}$. The Fourier-coefficients $\{c_f\}_{f=1}^{k}$ of t can efficiently be calculated in $\mathcal{O}(k \log k)$ time with the Fast Fourier Transform (FFT) algorithm. DFT captures the signal's periodic behavior by transforming the time series into the frequency domain. If different periodic behavior characterize the time series classes of the underlying application, it is worth to calculate e.g. the Euclidean distance of the Fourier-coefficients $\{c_f(t_1)\}_{f=1}^{k}$ and $\{c_f(t_2)\}_{f=1}^{k}$ of two time series t_1 and t_2: $d_{FE}(t_1, t_2) = \sqrt{\sum_{f=1}^{k}(c_f(t_1) - c_f(t_2))^2}$.

While DFT captures *global periodic* behavior, *wavelets* reflect both *local* and *global* character of a time series [7]. We use the recursive Haar Wavelet decomposition[1] of a time series t that results in a set of Wavelet-coefficients $\{w_i(t)\}_{i=1}^{k}$. Similarly to d_{FE}, we can calculate the Euclidean distance of these Wavelet-coefficients, denoted as d_{WE}.

In order to be able to capture further aspects of similarity, we use the following similarity measures (see [6] and the references therein for a more detailed description): (a) DISSIM that computes the similarity of time series with different sampling rates, (b) distance based on longest common subsequences (LCSS), (c) edit distance on real sequences (EDR), (d) edit distance with real penalty (EPR) that combines DTW and EDR.

4 Fusion of Similarity Measures

In the recent work of Ding et al. [6], none of the examined similarity measures could outperform DTW in general. However, in some specific tasks, one or the

[1] See http://www.ismll.uni-hildesheim.de/lehre/ip-08w/script/imageanalysis-2up-05-wavelets.pdf for an example.

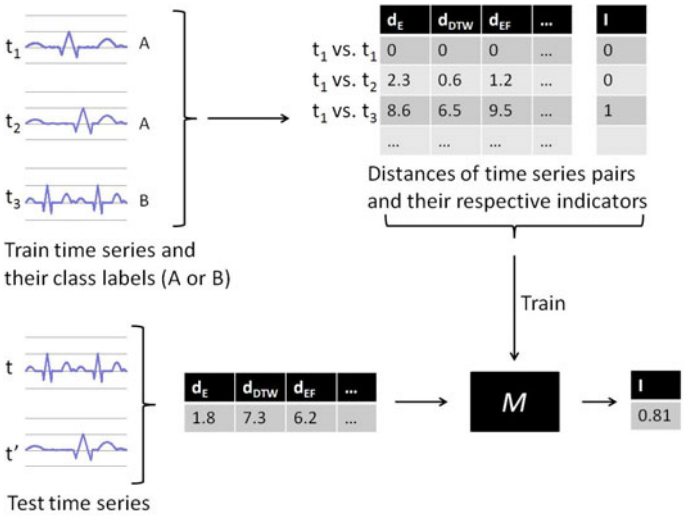


Fig. 1. Example: fusion of time series similarity measures. A regression model M is trained and its output is used as similarity measure.

other similarity measure worked better than DTW, which is likely to be explained by the fact that different aspects of similarity are relevant in different domains. In the case of simple tasks, one of the similarity measures may capture the relevant aspects of similarity entirely. This best similarity measure can be found based on domain knowledge or by measuring e.g. the leave-one-out classification error on the *train* data for the candidate similarity measures. In more complex cases, however, a single similarity measure may not be sufficient alone. Thus, we need to combine several ones. Such hybridization is often achieved in an ad hoc manner. In contrast, we develop a fusion schema for time series similarity measures that allows to combine similarity measures in a principled way.

In order to distinguish between the similarity measures that we want to combine and the resulting similarity measure, we refer to the former ones as *elementary similarity measures* whereas to the later one as *fused similarity measure*. Our approach for fusion of similarity measures[2] consists of the following steps, see also Figure 1 for an example:

1. For all the *pairs* of time series in the train data, we calculate the similarity values using all the considered elementary similarity measures.
2. In some of the above pairs, both time series belong to the same class, in others they belong to different classes. We define the indicator $\mathcal{I}(t_1, t_2)$ of a pair of time series (t_1, t_2), as follows: $\mathcal{I}(t_1, t_2) = 0$ if t_1 and t_2 belong to the same class, $\mathcal{I}(t_1, t_2) = 1$ otherwise.

[2] Note that we do *not* assume the elementary similarity measures to fulfill specific properties (such as triangular inequality).

3. We train a regression model $\mathcal{M}$. We use the similarity values (see first step) as training data along with the corresponding indicators as labels.
4. We propose to use the output of $\mathcal{M}$ as the fused similarity measure. For a pair of time series (t', t), where either or both of them can be unlabeled (test) time series, we calculate the similarity values using all the considered elementary similarity measures. Then use $\mathcal{M}$ to predict (based on these similarity values) the likelihood that t and t' belong to the different classes. Finally, we use this prediction as the distance of t and t'.

Note that our approach is generic, as this framework allows the fusion of arbitrary similarity measures using various regression models as $\mathcal{M}$. Furthermore, this fused similarity measure can be used by various classification algorithms.

Also note that the above description is just the conceptual description of our approach. While implementing it, one would not separately calculate the similarity of the pair (t_1, t_2) and (t_2, t_1) if the used elementary similarity measure is symmetric. Furthermore, one can pre-calculate and store the similarities of many pairs in case if the classification algorithm (which uses this fused similarity measure) queries the similarity of the same pair several times.

While fusing elementary similarity measures according to the above description, we consider all the pairs of time series. Therefore, if the training data contains n time series, the elementary similarity values are required to be calculated $\mathcal{O}(n^2)$ times and the data used to train $\mathcal{M}$ contains $\mathcal{O}(n^2)$ records. In case of small data sets, this is not a problem. For large datasets, we propose to sample the pairs, and calculate the elementary similarity values only for the sample. In this case, a large enough sample is sufficient for training $\mathcal{M}$.

As mentioned before, in simple domains, one single similarity measure might be sufficient to capture all the relevant aspects of similarity. In such cases, fusion of similarity measures is not necessary and could introduce noise. In order to avoid it, we propose to select the best similarity measure out of some *fused* similarity measures (with different regression models $\mathcal{M}$) *and* all the elementary similarity measures. In order to allow for this selection, we can judge the quality of each similarity measure by its leave-one-out nearest neighbor classification error on the train data.

5 Experiments

Datasets. We examined 35 out of all the 38 datasets used in [6]. We excluded 3 of them (Coffee, Beef, OliveOil) due to their tiny size (less than 100 time series).

Considered Similarity Measures. We used all the elementary similarity measures described in Section 3. We use two versions of DTW with warping window sizes constrained at 5% and 10% around the matrix diagonal [11] [12].

Comparison Protocol. As discussed in Section 1, 1-NN has been shown to be competitive and often even superior to many state-of-the art time series classification algorithms. Therefore, we compare time series similarity measures in

context of 1-NN classification. We measure classification error as the misclassification ratio. We perform 10-fold cross validation. For each dataset we test whether the differences between the performance of our approach and its competitors is statistically significant (t-test at significance level of 0.05).[3]

Baselines. We use two state-of-the art time series classifiers as baselines. The first one is the 1-NN using DTW with window size constrained at 5%. We denote it as DTW. For our second baseline, ELEM, we select the best *elementary* similarity measure based on the leave-one-out classification error on the train data and we use that similarity measure in the 1-NN classifier.

Fusion of Similarity Measures. We produce two fused similarity measures: as $\mathcal{M}_1$ and $\mathcal{M}_2$ we use (i) linear regression and (ii) multilayer perceptron[4] from the Weka machine learning library (http://www.cs.waikato.ac.nz/ml/weka/). After training $\mathcal{M}_1$ and $\mathcal{M}_2$, we select the best similarity measure out of the *fused* and *elementary* similarity measures based on the leave-one-out nearest neighbor classification error on the train data. Finally, we use the selected similarity measure in the 1-NN classifier. This approach is denoted as FUSION.

Results. For many of the examined datasets, the classification task is simple: DTW's error rates are less than 10 %. In these cases, all methods worked equally well, we did not observe statistically significant differences. For the remaining 22 non-trivial datasets, our results are shown in Tab. 1 and summarized in Tab. 2. In Tab. 1 bold font denotes the winner, in case of ties we use italic fonts. Whenever FUSION outperformed any of the baselines, we provide a symbol in form of $\pm/\pm$ where $+$ denotes significance and $-$ its absence against DTW and ELEM respectively. The baselines never outperformed FUSION significantly.[5] Note that we are not concerned with binary classification problems as the number of classes is more than two in most of the cases. In fact, this is one of the reasons why these datasets are challenging and this explains the relatively high error rates.

Discussion. As mentioned before, in simple domains an appropriately chosen elementary similarity measure can lead to very good classification accuracy, whereas a hybrid similarity measure, like the one we introduced, is necessary in more complex cases. Our experimental results show that based on the

[3] In order to save computational time, as discussed in Section 4, for some large datasets we randomly sample the pairs: we calculate similarities in case of Faces and Motes for 10 %, ChlorineConcentration for 5%, Mallat and TwoPatterns for 2%, Yoga and Wafer for 1 %, CinC and StarLightCurves for 0.5 % of all the pairs. In order to ensure fair comparison, we used the same sample of pairs both in our approach and for the baselines.

[4] We used Weka's standard parameter-settings, i.e. learning rate: 0.3, momentum: 0.2, number of train epochs: 500.

[5] We note that in 15 out of the 22 non-trivial datasets $\mathcal{M}_2$ (the similarity measure fused by multilayer perceptron) outperformed $\mathcal{M}_1$ (the similarity measure fused by linear regression). These datasets are: 50words, Adiac, Car, FacesUCR, Haptics, Lighting2, Lighting7, ChlorineConcentration, CinC, InlineSkate, Mallat, StarLightCurves, SwedishLeaf, WordsSynonyms, Yoga.

Table 1. Examined non-trivial datasets, their sizes and classification errors (in %)

Dataset	Size	DTW	ELEM	FUSION	Dataset	Size	DTW	ELEM	FUSION
Haptics	463	**55.9**	57.4	58.7	Star[d]	9236	23.2	20.8	**15.0**+/+
InlineSkate	650	51.4	**45.2**	46.2+/-	Lighting2	121	23.1	*20.6*	*20.6*
Chlorine[a]	4307	50.1	*47.3*	*47.3*+/-	Lighting7	143	**23.1**	25.1	25.1
Yoga	3300	42.5	41.8	**40.1**+/+	Words[e]	905	21.8	22.5	**21.4**-/-
Adiac	781	41.0	40.7	**35.2**+/+	ECG200	200	20.0	*14.5*	*14.5*+/-
Car	120	37.5	32.5	**28.3**+/-	Swedish[f]	1125	17.5	*11.5*	*11.5*+/-
OSULeaf	442	33.9	**21.7**	22.8+/-	FaceFour	112	16.0	**9.8**	11.6
TwoP[b]	5000	32.0	*0.2*	*0.2*+/-	CinC	1420	15.0	7.4	**4.3**+/+
FISH	350	26.6	**16.9**	17.4+/-	Motes	1272	14.0	*7.0*	*7.0*+/-
Medical[c]	1141	24.2	*23.4*	*23.4*-/-	Mallat	2400	11.6	11.6	**10.0**+/+
50words	905	23.4	24.2	**22.4**-/-	FacesUCR	2250	11.6	*7.7*	*7.7*+/-

[a]ChlorineConcentration, [b]TwoPatterns, [c]MedicalImages, [d]StarLightCurves, [e]WordsSynonyms, [f] SwedishLeaf.

Table 2. Number of FUSION's wins/loses and ties against DTW and ELEM

	against DTW		**against ELEM**	
	total	significant	total	significant
Wins	20	15	8	5
Ties	0	-	9	-
Loses	2	0	5	0

leave-one-out nearest neighbor classification error of the train data, FUSION could successfully identify those cases where the fusion of similarity measures is beneficial. Therefore, FUSION significantly outperformed DTW in 15 cases and ELEM in 5 cases, while FUSION never lost significantly against the baselines.

6 Conclusions and Outlook

Motivated by recent results, in this paper we focused on similarity measures for time series classification. We discussed what aspects of similarity they capture. As in complex applications several of these similarity aspects may be relevant simultaneously, we developed a generic framework which allowed fusion of various similarity measures. In our experiments over a large collection of real-world datasets, we showed that such complex applications exist and our approach achieved statistically significant improvements in those cases.

Our method for the fusion of similarity measures is not limited to time series classification. As future work, we would like to examine fusion of similarity measures in other contexts such as vector data or more complex, structured data. As our approach works on data instance pairs, for large datasets, we aim at exploring sampling strategies with special focus on the possibly imbalanced nature of the pair indicators. We would also like to examine in more depth, if *all* the

similarity measures are worth to be fused or one should rather select a subset of them, because many of them could do better (and hopefully faster) than all [20].

Acknowledgements. Research partially supported by the Hungarian National Research Fund (Grant Number OTKA 100238).

References

1. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning 36(1), 105–139 (1999)
2. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: Time-Series Classification based on Individualised Error Prediction. In: International Conference on Computational Science and Engineering. IEEE, Los Alamitos (2010)
3. Chan, K., Fu, A.: Efficient time series matching by wavelets. In: 15th International Conference on Data Engineering, pp. 126–133 IEEE, Los Alamitos (1999)
4. Christen, P.: Automatic record linkage using seeded nearest neighbour and support vector machine classification. In: Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 151–159. ACM Press, New York (2008)
5. Dietterich, T.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning 40(2), 139–157 (2000)
6. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proceedings of the VLDB Endowment 1(2), 1542–1552 (2008)
7. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference, and prediction, ch.5 Springer, Heidelberg (2009)
8. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Mining and Knowledge Discovery 7(4), 349–371 (2003)
9. Keogh, E., Shelton, C., Moerchen, F.: Workshop and challenge on time series classification (2007), http://www.cs.ucr.edu/~eamonn/SIGKDD2007TimeSeries.html
10. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping for datamining applications. In: 6th ACM SIGKDD Int'l. Conf. on Knowledge Discovery and Data Mining, pp. 285–289. ACM,New York (2000)
11. Ratanamahatana, C., Keogh, E.: Everything you know about dynamic time warping is wrong. In: SIGKDD Int'l. Wshp. on Mining Temporal and Seq. Data (2004)
12. Ratanamahatana, C., Keogh, E.: Making time-series classification more accurate using learned constraints. In: SIAM Int'l. Conf. on Data Mining, pp. 11–22 (2004)
13. Rath, T.M., Manmatha, R.: Word image matching using dynamic time warping. In: Conference on Computer Vision and Pattern Recognition, vol. 2. IEEE, Los Alamitos (2003)
14. Romano, L., Buza, K., Giuliano, C., Schmidt-Thieme, L.: XMedia: Web People Search by Clustering with Machinely Learned Similarity Measures. In: 18th WWW Conference on 2nd Web People Search Evaluation Workshop, WePS 2009 (2009)
15. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. Acoustics, Speech and Signal Processing 26(1), 43–49 (1978)
16. Schuller, B., Reiter, S., Muller, R., Al-Hames, M., Lang, M., Rigoll, G.: Speaker independent speech emotion recognition by ensemble classification (2005)

17. Sonnenburg, S., Ratsch, G., Schafer, C., Scholkopf, B.: Large scale multiple kernel learning. The Journal of Machine Learning Research 7, 1531–1565 (2006)
18. Ting, K., Witten, I.: Stacked generalization: when does it work?. In: 15th Int'l. Joint Conf. on Artifical Intelligence, vol. 2, pp. 866–871. Morgan Kaufmann, San Francisco (1997)
19. Zhang, G., Berardi, V.: Time series forecasting with neural network ensembles: an application for exchange rate prediction. Journal of the Operational Research Society 52(6), 652–664 (2001)
20. Zhou, Z., Wu, J., Tang, W.: Ensembling neural networks: Many could be better than all. Artificial intelligence 137(1-2), 239–263 (2002)

Enhancing IPADE Algorithm with a Different Individual Codification

Isaac Triguero[1], Salvador García[2], and Francisco Herrera[1]

[1] Dept. of Computer Science and Artificial Intelligence, CITIC-UGR
University of Granada. 18071, Granada, Spain
triguero@decsai.ugr.es, herrera@decsai.ugr.es
[2] Dept. of Computer Science. University of Jaén. 23071, Jaén, Spain
sglopez@ujaen.es

Abstract. Nearest neighbor is one of the most used techniques for performing classification tasks. However, its simplest version has several drawbacks, such as low efficiency, storage requirements and sensitivity to noise. Prototype generation is an appropriate process to alleviate these drawbacks that allows the fitting of a data set for nearest neighbor classification. In this work, we present an extension of our previous proposal called IPADE, a methodology to learn iteratively the positioning of prototypes using a differential evolution algorithm. In this extension, which we have called IPADECS, a complete solution is codified in each individual. The results are contrasted with non-parametrical statistical tests and show that our proposal outperforms previously proposed methods.

1 Introduction

The nearest neighbor (NN) algorithm [1] and its derivatives have been shown to perform well for classification problems. These algorithms are also known as instance-based learning and belong to the lazy learning family [2]. NN is a non-parametric classifier, which requires the storage of the entire training set and the classification of unseen cases, finding the class labels of the closest instances to them. The effectiveness of the NN may be affected with several weaknesses such as high computational cost, high storage requirement and sensitivity to noise. Furthermore, NN makes predictions over existing data and it assumes that input data perfectly delimits the decision boundaries among classes.

A successful technique which simultaneously tackles the computational complexity, storage requirements and sensitivity to noise of NN is based on data reduction. These techniques aim to obtain a representative training set with a lower size compared to the original one and with similar or even higher classification accuracy for new incoming data. Data reduction can be divided into two different approaches, known as prototype selection [3],[4] and Prototype Generation (PG) or abstraction [5]. The former process consists of choosing a subset of the original training data, while PG is not only able to select data, but can also build new artificial prototypes.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 262–270, 2011.

In the specialized literature, a great number of PG techniques have been proposed. Since the first approach PNN based on merging prototypes [6] and divide-and-conquer based schemes [7], many other proposals of PG can be found. For instance, ICPL [5] and RSP [8]. Positioning adjustment of prototypes is another perspective within the PG methodology. It aims to correct the position of a subset of prototypes from the initial set by using an optimization procedure. Many proposals belong to this family, such as learning vector quantization [9], particle swarm optimization [10] and differential evolution (DE) [11].

In [11] we proposed an iterative prototype adjustment procedure based on DE (IPADE) to automatically find the smallest reduced set with the appropriate number of instances for each class, which achieves a suitable classification accuracy over different types of problems. This method follows an incremental approach. At each step, an optimization procedure is used to adjust the positioning of the prototypes, and the method adds new prototypes if needed. We adopted the DE technique as optimizer.

In this work, we use a different individual codification for the DE algorithm to increase the optimization capabilities. Concretely, this proposal follows an IPADE approach with a complete solution per individual (IPADECS). In experiments on 20 real-world benchmark data sets, the classification accuracy and reduction rate of our approach are investigated and its performance will be compared with IPADE, which showed to outperform previously proposed methods.

In order to organize this paper, Section 2 describes the background of PG and DE. Section 3 explains the IPADE algorithm. Section 4 shows the IPADECS approach. Section 5 discusses the experimental framework and presents the analysis of results. Finally, in Section 6 we summarize our conclusions.

2 Background

This section covers the background information necessary to define our proposal. Subsection 2.1 presents the background on PG. Next, Subsection 2.2 shows the main characteristics of DE.

2.1 Prototype Generation

PG can be defined as the application of instance construction algorithms over a data set to improve the classification accuracy of a NN classifier. Specifically, PG can be defined as follows: Let $\mathbf{x}_p$ be an instance where $\mathbf{x}_p = (\mathbf{x}_{p1}, \mathbf{x}_{p2}, ..., \mathbf{x}_{pm}, \mathbf{x}_{p\omega})$, with $\mathbf{x}_p$ belonging to a class ω of Ω possible classes given by $\mathbf{x}_{p\omega}$ and a m-dimensional space in which $\mathbf{x}_{pi}$ is the value of the i-th feature of the p-th sample. Furthermore, let $\mathbf{x}_t$ be an instance where $\mathbf{x}_t = (\mathbf{x}_{t1}, \mathbf{x}_{t2}, ..., \mathbf{x}_{tm}, \mathbf{x}_{t\psi})$, with $\mathbf{x}_t$ belonging to a class ψ, that it is unknown, of Ω possible classes. Then, let us assume that there is a training set TR which consists of n instances $\mathbf{x}_p$ and a test set TS composed by s instances $\mathbf{x}_t$. The purpose of PG is to obtain a prototype generated set (PGS), which consists of r, $r < n$, prototypes $\mathbf{p}_u$ where $\mathbf{p}_u = (\mathbf{p}_{u1}, \mathbf{p}_{u2}, ..., \mathbf{p}_{um}, \mathbf{p}_{u\omega})$, which are generated from the examples of TR. PGS must represent efficiently the distributions of the classes.

2.2 Differential Evolution

DE starts with a population of NP solutions, so-called individuals. The generations are denoted by $G = 0, 1, \ldots, G_{max}$. It is usually to denote each individual as a D-dimensional vector $X_{i,G} = \{x^1_{i,G},..., x^D_{i,G}\}$, called a "target vector".

After initialization, DE applies the mutation operator to generate a mutant vector $V_{i,G}$, with respect to each individual $X_{i,G}$, in the current population. For each target $X_{i,G}$, at the generation G, its associated mutant vector $V_{i,G}$= $\{V^1_{i,G},..., V^D_{i,G}\}$. In this contribution, we focus on the *RandToBest/1* which generates the mutant vector as follows:

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r^i_1,G} - X_{r^i_2,G}) \quad (1)$$

The indices r^i_1, r^i_2 are mutually exclusive integers randomly generated within the range $[1, NP]$, which are also different from the base index i. The scaling factor F is a positive control parameter for scaling the different vectors.

After the mutation phase, the crossover operation is applied to each pair of the target vector $X_{i,G}$ and its corresponding mutant vector $V_{i,G}$ to generate a new trial vector that we denote $U_{i,G}$. We will focus on the binomial crossover scheme, which is performed on each component whenever a randomly picked number between 0 and 1 is less than or equal to the crossover rate (CR), which controls the fraction of parameter values copied from the mutant vector. Then, we must decide which individual should survive in the next generation $G + 1$. If the new trial vector yields an equal or better solution than the target vector, it replaces the corresponding target vector in the next generation; otherwise the target is retained in the population.

The success of DE in solving a problem crucially depends on the associated control parameter values (F and CR) that determine the convergence speed. One of the most successful adaptive DE algorithms is SFLSDE [12], we use the ideas established in this work to obtain a self adaptive algorithm.

3 Iterative Prototype Adjustment Based on Differential Evolution

In this section, we describe the IPADE approach. IPADE follows an iterative scheme, in which it determines the most appropriate number of prototypes per class and their best positioning. Concretely, IPADE is divided into three different stages: initialization (Subsection 3.1), optimization (Subsection 3.2) and addition of prototypes (Subsection 3.3). Figure 1 shows the pseudocode. In the following we describe the most significant instructions enumerated from 1 to 26.

3.1 Initialization

A random selection (stratified or not) of examples from TR may not be the most adequate procedure to initialize the PGS. Instead, IPADE iteratively learns prototypes in order to find the most appropriate structure of PGS. Instruction 1

```
1: PGS = Initialization(TR)
2: DE_Optimization(PGS, TR)
3: Accuracy_Global = Evaluate(PGS, TR)
4: registerClass[0..Ω] = optimizable
5: while Accuracy_Global <> 1.0 or all classes are non − optimizables do
6:    lessAccuracy = ∞
7:    for i = 1 to Ω do
8:       if registerClass[i] == optimizable then
9:          AccuracyClass[i] = Evaluate (PGS, Examples of class i in TR)
10:         if AccuracyClass[i] < lessAccuracy then
11:            lessAccuracy = AccuracyClass[i]
12:            targetClass = i
13:         end if
14:      end if
15:   end for
16:   PGS_test = PGS ∪ RandomExampleForClass(TR, targetClass)
17:   DE_Optimization(PGS_test, TR)
18:   accuracy_Test = Evaluate(PGS_test, TR)
19:   if accuracy_Test > Accuracy_Global then
20:      Accuracy_Global = accuracy_Test
21:      PGS = PGS_test
22:   else
23:      registerClass[targetClass] = non − optimizable
24:   end if
25: end while
26: return PGS
```

Fig. 1. IPADE algorithm basic structure

generates the initial solution PGS. In this step, PGS represents each class distribution with its respective centroid. The centroid of the class does not completely cover the region of each class and it does not avoid misclassifications. Thus, instruction 2 applies the first optimization stage using the initial PGS composed of centroids for each class. The optimization stage must modify the prototypes of PGS using the movement idea in the m-dimensional space. It is important to point out that we normalize all attributes of the data set to the $[0, 1]$ range.

3.2 Differential Evolution Optimization for IPADE

In this subsection we explain the proposal to apply the underlying idea of the DE algorithm to the PG problem as a position adjusting of prototypes scheme. In the proposed IPADE algorithm, each individual in the population encodes a single prototype without the class label and, as such, the dimension of the individuals is equal to the number of attributes of the specific problem. An individual classifies an example of TR when it is the closest particle to that example.

The DE algorithm uses each prototype $\mathbf{p}_u$ of PGS, provided by the IPADE algorithm, as an initial population. Next, mutation and crossover operators guide the optimization of the positioning of each $\mathbf{p}_u$. These operators only produce modifications in the attributes of the prototypes of PGS. Hence, the class value remains unchangeable throughout the evolutionary cycle. IPADE focuses on the *DE/CurrentToRand/1* strategy to generate the trial prototypes $\mathbf{p}_u^{'}$.

After this, we obtain a trial solution $PGS^{'}$, which is constituted for each $\mathbf{p}_u^{'}$. The selection operator decides which solution $PGS^{'}$ or PGS should survive for the next iteration. The NN rule guides this operator to obtain the corresponding fitness value. Implementation of these operators was described in depth in [11].

Instruction 3 evaluates the accuracy of the initial solution, measured by classifying the examples of TR with the prototypes of GS by using the NN rule.

3.3 Addition of Prototypes

After the first optimization process, IPADE enters in an iterative loop (Instructions 5-25) to determine which classes need more prototypes to faithfully represent their class distribution. In order to do this, we need to define two types of classes. A class ω is said to be optimizable if it allows the addition of new prototypes to improve its local classification accuracy. The local accuracy of ω is computed by classifying the examples of TR whose class is ω with the prototypes kept in PGS (using the NN rule). The *target class* will be the *optimizable* class with the least accuracy registered. From instructions 7 to 15, the algorithm identifies the *target class* in each iteration. All classes start as *optimizable*.

In order to reduce the classification error of the *target class*, IPADE extracts a random example of this class from TR and adds this to the current PGS in a new trial set PGS_{test} (Instruction 16). This addition forces the re-positioning of the prototypes of PGS_{test} by again using the optimization process (Instruction 17) and its corresponding evaluation (Instruction 18) of predictive accuracy.

After this process, we have to ensure that the new positioning of prototypes of PGS_{test} has reported a successful improvement of the accuracy rate in respect to the previous PGS. If the global accuracy of the PGS_{test} is lesser than the accuracy of PGS, IPADE does not add this prototype to PGS and this class is registered as *non-optimizable*. Otherwise, $PGS = PGS_{test}$.

The stopping criterion is satisfied when the accuracy rate is 1.0 or all the classes are registered as *non-optimizable*. The algorithm returns PGS as the smallest reduced set which is able to classify the TR appropriately.

4 IPADECS: IPADE with a Complete Solution Per Individual

In this work we choose a different codification scheme for the DE optimization. Concretely, a complete solution is codified in each individual. This corresponds to the case where each individual of the population encodes a complete PGS. Following the notation used in Subsection 2.2, $X_{i,G}$ defines the target vector, but in this case, the target vector can be represented as a matrix. Table 1 describes the structure of an individual. Furthermore, each prototype p_j, $1 \leq j \leq \mathbf{r}$, of an individual $X_{i,G}$ has a class $x_{p\omega,j}$.

The current PGS of the IPADE algorithm must be inserted once as one of the individuals of the population, initializing the rest of the individuals with random prototypes extracted from TR. It is important to point out that every solution must have the same structure that PGS, thus they must have the same number of prototypes per class, and the classes must have the same arrangement in the matrix $X_{i,G}$.

Table 1. Encoding of a set of prototypes in a individual $X_{i,G}$

	Attribute 1	**Attribute 2**	...	**Attribute** D	**Class**
Prototype 1	$\mathbf{x}_{p1,1}$	$\mathbf{x}_{p2,1}$	...	$\mathbf{x}_{pD,1}$	$\mathbf{x}_{p\omega,1}$
Prototype 2	$\mathbf{x}_{p1,2}$	$\mathbf{x}_{p2,2}$	...	$\mathbf{x}_{pD,2}$	$\mathbf{x}_{p\omega,2}$
...					
Prototype r	$\mathbf{x}_{p1,r}$	$\mathbf{x}_{p2,r}$	...	$\mathbf{x}_{pD,r}$	$\mathbf{x}_{p\omega,r}$

The *RandToBest* mutation strategy generates the mutant matrix $V_{i,G}$ in respect to each individual $X_{i,G}$, in the current population. The operations of addition, subtraction and scalar product are carried out as typical matrices. This is the justification for the individuals having the same structure. In order for the mutation operator to make sense, the operators must act over the same attributes and over prototypes of the same class in all cases. After applying this operator, we check that the mutant matrix $V_{i,G}$ has generated correct values for all features of the prototypes. This procedure checks if there have been values out of range of $[0, 1]$. If a computed value is greater than 1, we truncate it to 1, and if is lower than 0, we establish it at 0.

The trial matrix $U_{i,G}$ is generated with a binomial crossover. In PG, the mutant matrix $V_{i,G}$ exchanges its prototypes with $X_{i,G}$ to generate $U_{i,G}$.

The selection operator decides which individual between $X_{i,G}$ and $U_{i,G}$ should survive in the next generation $G + 1$. The NN rule, with k=1 (1NN), guides this operator. The instances in TR are classified with the prototypes encoded in $X_{i,G}$ or $U_{i,G}$ by the 1NN rule with a *leave-one-out* validation scheme, and their corresponding fitness values are measured as the number of successful hits relative to the total number of classifications. We try to maximize this value.

5 Experimental Framework and Analysis of Results

This section presents the experimental framework (Subsection 5.1) and the comparative study between IPADECS and IPADE (Subsection 5.2).

5.1 Experimental Framework

In this section we show the issues related to the experimental study. In order to compare the performance of the algorithms, we use the *accuracy* rate [14], and the *reduction rate* measured as: $1 - size(PGS)/size(TR)$.

We use 20 data sets from the KEEL-dataset repository[1] [13]. Table 2 shows, for each data set, the number of examples (#Ex.), the number of attributes (#Atts.), and the number of classes (#Cl.). The data sets considered are partitioned using the ten fold cross-validation (10-fcv) procedure.

The configuration parameters of all the methods used in the comparison are shown in Table 3. In this table, the values of the parameters F_l, F_u, $iterSFGSS$

[1] http://sci2s.ugr.es/keel/datasets

Table 2. Summary description for used classification data sets

Data Set	#Ex.	#Atts.	#Cl.	Data Set	#Ex.	#Atts.	#Cl.
bupa	345	6	2	nursery	12690	8	5
car	1.728	6	4	pageblocks	5.472	10	5
ecoli	336	7	8	pima	768	8	2
german	1.000	20	2	ring	7400	20	2
glass	214	9	7	segment	2.310	19	7
haberman	306	3	2	thyroid	7.200	21	3
heart	270	13	2	twonorm	7.400	20	2
iris	150	4	3	wine	178	13	3
monks	432	6	2	wisconsin	683	9	2
newthyroid	215	5	3	zoo	101	17	7

Table 3. Parameter specification for the methods employed in the experimentation

Algorithm	*Parameters*
IPADE	iterations of Basic DE = 500, iterSFGSS = 8, iterSFHC =20, Fl=0.1, Fu=0.9
IPADECS	PopulationSize = 10, iterations of Basic DE = 500 iterSFGSS = 8, iterSFHC = 20, Fl=0.1, Fu=0.9

and $iterSFHC$ are the recommended values established in [12]. Furthermore, euclidean distance is used as a similarity function and those which are stochastic methods have been run three times per partition. Implementations of the algorithms can be found in the KEEL software tool [13].

5.2 Analysis of Results

Table 4 shows the results obtained in test data. This table collects the average results in terms of accuracy and reduction rate, obtained over the 20 data sets considered. The best result for each data set is remarked in bold.

Considering only average results could lead us to erroneous conclusions. Due to this fact, we will accomplish statistical comparisons over multiple data sets based on non-parametric tests [15], [16]. Specifically, we use the Wilcoxon Signed-Ranks test and we obtain a p-value of **0.0019** in the comparison between IPADECS and IPADE.

Looking at Table 4 we want to outline some interesting comments:

- IPADECS obtains the best average results in terms of accuracy in comparison with the IPADE algorithm. Furthermore, the Wilcoxon test supports this statement with a p-value lower than 0.05.
- As we can observe in Table 4, the original IPADE achieves a higher reduction rate than IPADECS. This behavior arises because IPADECS uses a stronger optimization process that allows to introduce more prototypes in PGS, specifically in the addition stage to increase the accuracy capabilities.

Table 4. Results obtained

Datasets	IPADE		IPADECS	
	AccTest	Red. Rate	AccTest	Red. Rate
bupa	0.5775±0.0545	0.9894±0.0021	**0.6567±0.0848**	0.9842±0.0055
car	0.8096±0.0298	0.9931±0.0012	**0.8692±0.0205**	0.9914±0.0021
ecoli	0.7714±0.0615	0.9683±0.0026	**0.7892±0.0654**	0.9524±0.0042
german	0.7160±0.0233	0.9966±0.0008	**0.7180±0.0325**	0.9920±0.0032
glass	0.6306±0.1092	0.9533±0.0053	**0.6909±0.1113**	0.9393±0.0131
haberman	0.7287±0.0715	0.9898±0.0027	**0.7445±0.0640**	0.9891±0.0049
heart	0.8185±0.0560	0.9877±0.0052	**0.8370±0.0983**	0.9831±0.0059
iris	**0.9667±0.0447**	0.9763±0.0030	0.9467±0.0400	0.9748±0.0036
monks	0.8576±0.0974	0.9920±0.0024	**0.9120±0.0476**	0.9910±0.0021
newthyroid	0.9630±0.0348	0.9783±0.0045	**0.9818±0.0302**	0.9835±0.0021
nursery	0.5299±0.0282	0.9993±0.0002	**0.6479±0.0458**	0.9992±0.0003
page-blocks	0.9291±0.0084	0.9983±0.0002	**0.9335±0.0155**	0.9974±0.0005
pima	0.7528±0.0586	0.9955±0.0010	**0.7684±0.0467**	0.9916±0.0031
ring	0.8165±0.0342	0.9989±0.0004	**0.8970±0.0103**	0.9956±0.0012
segment	0.8753±0.0142	0.9950±0.0009	**0.9208±0.0097**	0.9919±0.0017
thyroid	**0.9404±0.0036**	0.9990±0.0002	0.9399±0.0036	0.9989±0.0003
twonorm	0.9764±0.0070	0.9995±0.0002	**0.9766±0.0069**	0.9993±0.0002
wine	**0.9497±0.0678**	0.9800±0.0025	0.9441±0.0352	0.9794±0.0040
wisconsin	**0.9699±0.0197**	0.9960±0.0011	0.9642±0.0194	0.9951±0.0019
zoo	0.9464±0.0702	0.9087±0.0071	**0.9633±0.0823**	0.9086±0.0054
AVERAGE	0.8263±0.1370	0.9848±0.0216	**0.8551±0.1138**	0.9819±0.0231

6 Concluding Remarks

In this contribution, we have presented a data reduction technique called IPADECS which iteratively learns the most adequate number of prototypes per class and their positioning for the NN algorithm. This technique is an extension our previous proposal IPADE with a different codification scheme. The results obtained show that IPADECS is statistically better than IPADE, which showed to outperform previously proposed methods.

Acknowledgments. This work was supported by Project TIN2008-06681-C06-01. I. Triguero holds an scholarship from the University of Granada.

References

1. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13, 21–27 (1967)
2. Garcia, E.K., Feldman, S., Gupta, M.R., Srivastava, S.: Completely Lazy Learning. IEEE Transactions on Knowledge and Data Engineering 22(9), 1274–1285 (2010)
3. Liu, H., Motoda, H. (eds.): Instance Selection and Construction for Data Mining. The International Series in Engineering and Computer Science. Springer, Heidelberg (2001)
4. García, S., Cano, J.R., Herrera, F.: A memetic algorithm for evolutionary prototype selection: A scaling up approach. Pattern Recognition 41(8), 2693–2709 (2008)
5. Lam, W., Keung, C., Liu, D.: Discovering Useful Concept Prototypes for Classification Based on Filtering and Abstraction. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(8), 1075–1090 (2002)

6. Chang, C.: Finding Prototypes For Nearest Neighbor Classifiers. IEEE Transactions on Computers 23(11), 1179–1184 (1974)
7. Chen, C.H., Jóźwik, A.: A sample set condensation algorithm for the class sensitive artificial neural network. Pattern Recognition Letters 17(8), 819–823 (1996)
8. Sánchez, J.S.: High training set size reduction by space partitioning and prototype abstraction. Pattern Recognition 37(7), 1561–1564 (2004)
9. Kohonen, T.: The self organizing map. Proceedings of the IEEE 78(9), 1464–1480 (1990)
10. Cervantes, A., Galván, G., Isasi, I.M.: AMPSO: A New Particle Swarm Method for Nearest Neighborhood Classification. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 39(5), 1082–1091 (2009)
11. Triguero, I., García, S., Herrera, F.: IPADE: Iterative Prototype Adjustment for Nearest Neighbor Classification. IEEE Transactions on Neural Networks 21(12), 1984–1990 (2010)
12. Ferrante, N., Tirronen, V.: Scale factor local search in differential evolution. Memetic Computing 1(2), 153–171 (2009)
13. 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. Journal of Multiple-Valued Logic and Soft Computing (2010) (in press)
14. Alpaydin, E.: Introduction to Machine Learning, 2nd edn. The MIT Press, Cambridge (2010)
15. García, S., Herrera, F.: An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. Journal of Machine Learning Research 9, 2677–2694 (2008)
16. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental Analysis of Power. Information Sciences 180, 2044–2064 (2010)

A Multi-objective Evolutionary Approach for Subgroup Discovery*

Victoria Pachón, Jacinto Mata, Juan Luis Domínguez, and Manuel J. Maña

Escuela Técnica Superior de Ingeniería. Universidad de Huelva, Carretera Palos de la Frontera SN. Huelva, Spain
{vpachon,mata,juan.dominguez,manuel.mana}@dti.uhu.es

Abstract. In this paper a new evolutionary multi-objective algorithm (GAR-SD) for Subgroup Discovery tasks is presented. This algorithm can work with both discrete and continuous attributes without the need for a previous discretization. An experimental study was carried out to verify the performance of the method. GAR-SD was compared with other subgroup discovery methods by evaluating certain measures (such as number of rules, number of attributes, significance, support and confidence). For Subgroup Discovery tasks, GAR-SD obtained good results compared with existing algorithms.

Keywords: Data Mining, Subgroup Discovery, Evolutionary Algorithms.

1 Introduction

In general, Data Mining techniques are usually divided into predictive and descriptive techniques. In predictive techniques, the models are induced from data labelled with a class and the aim is to predict the value of the class for instances in which the class is previously unknown. On the other hand, the main objective of the descriptive techniques is to find understandable and useful patterns, based on unlabelled data.

A new emerging task consists of obtaining descriptive knowledge regarding a property of interest, in other words, to discover significant differences between groups that exist in a dataset and which can be interpreted by a human. In recent years, three approaches have aroused the interest of researchers: *Contrast Set Mining* [1], *Emerging Pattern Mining* [2] and *Subgroup Discovery* [3][4]. These techniques are known as *Supervised Descriptive Rule Discovery*. In [5], a unifying survey of these techniques is presented. This paper focuses on the *Subgroup Discovery* (*SD*) task, whose goal is the discovery of interesting patterns in relation to a specific property of interest for the user.

In recent years, several SD algorithms have been presented. Some of them are adaptations of classic deterministic algorithms for rule induction as in [6][7], where

* This work was partially funded by the Spanish Ministry of Science and Innovation, the Spanish Government Plan E and the European Union through ERDF (TIN2009-14057-C03-03).

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 271–278, 2011.

CN2-SD and APRIORI-SD algorithms are developed. One algorithm for exhaustive subgroup discovery based on the FP-growth algorithm is the SD-Map method [8]. There are some works [9][10] where algorithms for SD in relational space are described. Other subgroup discovery algorithms were developed implementing genetic fuzzy systems [11] to solve subgroup discovery tasks. In this group, the algorithms are designated SDIGA [12], MESDIF [13] and NMEEF-SD [14].

As in the descriptive tasks, the handling of numerical attributes is a great problem in the discovery of the most significant rules. Most of the aforementioned algorithms perform a prior discretization of the numeric attributes before carrying out the rule induction process. For example, APRIORI-SD and CN2-SD discretize the numeric attributes by using the Fayyad discretization method [15] and in the genetic fuzzy systems the authors use a fixed number of linguistic labels for the numeric attributes.

This paper presents a multi-objective evolutionary approach to find the most important rules of the subgroups on various quality measures. In this approach, GAR-SD optimizes support, confidence and significance. The major contribution of GAR-SD is that it does not carry out a discretization of numeric attributes before the rule induction process. The ranges of numeric attributes are obtained in the rules induction process itself. In this way, it is ensured that these intervals are the most suitable for maximising the quality measures. The GAR-SD algorithm is based on the GAR algorithm [16], which was developed to find association rules. To verify the validity of the method presented, a comparative study was performed for evolutionary SD algorithms SDIGA, MESDIF and NMEEF-SD, as well as the classic SD algorithms CN2-SD and APRIORI-SD.

The remainder of the paper is structured as follows: the next section gives a short description of SD. In Section 3, the new evolutionary approach to obtain rules for SD is explained. In Section 4, the experimental study is presented and the results obtained are analyzed. Finally, in Section 5 the conclusions and further research are presented.

2 Subgroup Discovery

As was defined in [5], "subgroup descriptions are conjunctions of features that are characteristic for a selected class of individuals (property of interest). A subgroup description can be seen as the condition part of a rule *SubgroupDescription* → *Class*. Therefore, subgroup discovery can be seen as a special case of a more general rule learning task."

One of the most important aspects of the subgroup discovery task is the selection of the measures to use. The most common in the literature are coverage, significance, unusualness, support and confidence. This study focuses on significance, support and confidence. These measures are defined as:

- **Significance (SIG):** Average of the likelihood ratio of each rule. Significance indicates how significant a finding is [6].
- **Support (SUP):** Percentage of target examples positives covered by the rules, computed as the true positive rate for the union of subgroups [6].
- **Confidence (CONF):** Average of the relative frequency of examples satisfying the complete rule among those satisfying only the antecedent.

Therefore, in this study other measures were considered, such as number of rules (*#Rul*, average of the number of inducted rules) and number of variables (*#Var*, average of the number of attributes of the antecedent).

3 GAR-SD Algorithm. Description of the Method

The GAR-SD algorithm is based on the theory of evolutionary algorithms [17]. It is designed to work with both numeric and discrete attributes (and, of course, both at once). The intervals are obtained in the evolutionary process itself, during the learning stage. The heart of the tool is a multi-objective evolutionary algorithm whose purpose is to find the most interesting rules for each subgroup that can be extracted from a database with both numeric and discrete attributes. The GAR-SD operating schema can be seen in Fig. 1.

```
input: database D
output: subgroup rules
Algorithm GAR-SD
   for each class
      nRules ← 0
      while (nrows not covered < S_SUP)
      nGen ← 0
      generate the initial population P(0)
           while (nGen < G) do
               evaluate individuals of P(nGen)
               P(nGen+1) ← Select individual of P(nGen)
               complete population of P(nGen+1)
                  through crossbreeding
               perform mutations in P(nGen+1)
               nGen = nGen+1
           end while
           R[nRules] ← select best individual of P(nGen)
           adjust bounds of R[nRules]
           nRules ← nRules + 1
      end while
   end for
end GAR-SD
```

Fig. 1. The GAR-SD algorithm

S_SUP is a predefined number indicating the percentage of examples covered by the rules of the subgroup. This is the stop condition. To facilitate the process of searching for different rules, the initial population is constructed according to examples in the dataset that have not been covered by rules found previously. The aim of this is to prevent all the populations finding the same rule.

3.1 Chromosome Representation and Initial Population

An individual corresponds to a rule of a subgroup, in which each gene represents the maximum and minimum values for the intervals of each numeric attribute forming the rule, or the set of values which each discrete attribute can take in the rule. The rules can have a variable number of attributes (continuous or discrete).

The initial population is generated by randomly generating the limits of the intervals for each of the continuous attributes, and additionally assigning an initial value to the discrete attributes forming the rule. In order to stimulate convergence, the individuals of the first population are conditioned to cover at least one example in the database. In the case of the continuous attributes, these start with a reduced interval size, while the discrete attributes start with a single value. The attributes of the rule are selected randomly.

3.2 Genetic Operators

The genetic operators used are the usual ones in evolutionary processes, that is, crossover and mutation. For replication, a percentage of the best individuals are selected. The rest of the population is completed by means of the crossover operator, randomly selecting the individuals to be combined to form new ones. A variation of the uniform crossover has been adopted: for each crossover between two individuals, two new ones are generated, and the one which best adapts passes on to the next generation. In Fig. 2, an example of crossover between two individuals (I_x and I_y) is showed.

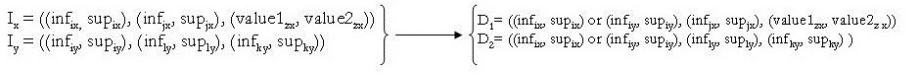


Fig. 2. Example of crossover

It was observed that in the solution to our algorithm, the mutation operator is perhaps the most influential during the evolutionary process. The process consists of altering one or more genes of the individual. A distinction is made between continuous and discrete type genes. In attributes of a continuous type, the mutation causes changes in the upper and lower limits of the interval. The discrete attribute mutation consists of changing the value of the attribute for another belonging to its domain, or adding or taking away values from the list of the attribute in question. At the end of the process, there is an adjustment or refinement to the limits of the intervals of the rule which has been found. This adjustment consists of reducing the size of the intervals in such a way that both limits are values which belong to the database in question.

3.3 Fitness Function

As described in Section 2, there are many measures in SD to obtain the quality of a subgroup. The main goal of the fitness function is to determine the best adapted individuals. In addition, other factors capable of establishing priority between individuals were considered, as explained below. The evaluation function which should be maximized for each individual *i* is given by (1).

Sup, *conf* and *sig* are related to support, confidence and significance of an individual (rule). Each parameter has an associated weighting factor. The rest of the additional parameters are described as follow:

- **AmplitudeC:** Average amplitude of all the numeric intervals making up the rule.
- **AmplitudeD:** Average amplitude of all discrete intervals forming the rule.
- **Re-covering:** This parameter aims to ensure that the algorithm tends to look for rules with different attributes in each new search.

$$f(i)= (sup*w_s) + (conf*w_c) + (sig*w_{sig}) - (amplitudeC*w_{ac}) - (amplitudeD*w_{ad}) - (re\text{-}covering*w_{recov}) \quad (1)$$

4 Experimental Evaluation

An experimental study with some of the datasets available in the UCI repository was carried out to analyze the behaviour of the proposed algorithm. Table 1 summarizes the main characteristics (number of attributes, number of discrete attributes, number of continuous attributes, number of classes and number of examples) of each selected dataset.

Table 1. Dataset characteristics

Dataset	#Attr	#Discr	#Cont	#Class	#Examples
Appendicitis	7	0	7	2	106
Diabetes	8	0	8	2	768
German	20	13	7	2	1000
Hypothyroid	25	18	7	2	3163
Iris	4	0	4	3	150

Table 2. Experimentation results for Appendicitis dataset

#Algorithm	#Rul	#Var	SIG	SUP	CONF
APRIORI-SD	2	2.25	2.051	0.828	0.811
CN2-SD	13	1.846	2.894	**0.971**	0.458
MESDIF-SD (3)	6	4.166	2.120	0.771	0.564
MESDIF-SD (5)	6	3.333	1.761	0.142	0.296
NMEEF-SD (3)	1	2	0.771	0.685	0.888
NMEEF-SD (5)	3	2	1.417	0.685	0.750
SDIGA-SD (3)	2	3.5	2.685	0.771	0.711
SDIGA-SD (5)	4	2.25	1.725	0.8	0.803
GAR-SD	4	3.25	**3.233**	0.545	**0.926**

To compare the algorithms, each dataset was divided into 2 parts. 60% of the examples were used to train the models and 40% were used as test set. In addition, for non-deterministic algorithms 5 runs were performed and averages of each measure were calculated. The rules obtained by GAR-SD were compared with those obtained by the APRIORI-SD, CN2-SD, NMEEF-SD and MESDIF-SD. For the execution of these algorithms, the implementations of that methods included in KEEL[18] were used. For classic algorithms APRIORI-SD and CN2-SD, a previous discretization of

numeric attributes was carried out using the Fayyad method. For genetic fuzzy systems, tests were conducted with 3 and 5 linguistic labels.

The experimental results are shown in Tables 2 to 6. Considering the different characteristics of the selected sets, the results obtained from these experiments show that GAR-SD performs better significance values than other algorithms. In addition, as GAR-SD is a multi-objective algorithm, good values of support, confidence, number of rules and number of variables are obtained.

Table 3. Experimentation results for Diabetes dataset

#Algorithm	#Rul	#Var	SIG	SUP	CONF
APRIORI-SD	7	2.142	13.804	0.490	0.750
CN2-SD	11	2	14.303	**0.961**	0.602
MESDIF-SD (3)	6	4	1.325	0.597	0.198
MESDIF-SD (5)	6	2.833	3.463	0.390	0.549
NMEEF-SD (3)	9	3.111	7.886	0.632	0.715
NMEEF-SD (5)	4	2	6.693	0.628	0.718
SDIGA-SD (3)	4	3.75	5.417	0.699	0.484
SDIGA-SD (5)	8	4.125	2.558	0.574	0.185
GAR-SD	12.5	3.016	**21.157**	0.701	**0.813**

Table 4. Experimentation results for German dataset

#Algorithm	#Rul	#Var	SIG	SUP	CONF
APRIORI-SD	5	2.4	16.088	0.396	0.648
CN2-SD	12	5.083	11.898	0.834	0.395
MESDIF-SD (3)	6	10.833	0.119	0.003	0.166
MESDIF-SD (5)	6	8.333	0.694	0.003	0.055
NMEEF-SD (3)	10	2.8	8.037	0.702	**0.775**
NMEEF-SD (5)	8	2.375	5.300	0.699	0.762
SDIGA-SD (3)	3	2	2.811	0.777	0.578
SDIGA-SD (5)	3	2	0.279	**1**	0.574
GAR-SD	9	3.6	**19.153**	0.94	0.614

Table 5. Experimentation results for Hypothyroid dataset

#Algorithm	#Rul	#Var	SIG	SUP	CONF
APRIORI-SD	-	-	-	**-**	**-**
CN2-SD	2	3	126.386	0.959	0.776
MESDIF-SD (3)	6	9.666	13.654	0.126	0.846
MESDIF-SD (5)	6	10.833	8.628	0.163	0.497
NMEEF-SD (3)	3	2.333	22.865	0.646	**0.984**
NMEEF-SD (5)	4	2.25	44.415	0.723	0.989
SDIGA-SD (3)	2	2	4.952	0.175	0.525
SDIGA-SD (5)	3	2	13.492	0.396	0.665
GAR-SD	9	5,55	**136.231**	**0.966**	0.7

Table 6. Experimentation results for Iris dataset

#Algorithm	#Rul	#Var	SIG	SUP	CONF
APRIORI-SD	15	1.6	24.78	**0.98**	0.932
CN2-SD	3	1	24.425	0.98	0.896
MESDIF-SD (3)	9	3.444	17.741	0.88	0.65
MESDIF-SD (5)	9	2.444	20.774	0.88	0.938
NMEEF-SD (3)	6	2.333	29.990	0.94	**0.975**
NMEEF-SD (5)	4	2	28.650	0.88	0.891
SDIGA-SD (3)	3	2.666	28.691	0.88	0.95
SDIGA-SD (5)	4	2.25	21.407	0.8	0.925
GAR-SD	2	2.333	**30.349**	0.941	0.958

Table 7 shows the average values of each measure for each of the algorithms. As can be seen, the average values of GAR-SD in significance are greater than those obtained by the other algorithms tested, while confidence and support are among the best results. As GAR-SD is a multi-objective algorithm, it tries to find rules in which these measures are maximized to an acceptable level. It was concluded that the subgroups obtained by GAR-SD are therefore good, useful and representative.

Table 7. Summary of results obtained

#Algorithm	#Rul	#Var	SIG	SUP	CONF
APRIORI-SD	7.25	2.098	14.18	0.673	0.785
CN2-SD	8.2	2.585	35.981	**0.941**	0.625
MESDIF-SD (3)	6.6	6.422	6.992	0.528	0.485
MESDIF-SD (5)	6.6	5.555	7.064	0.431	0.467
NMEEF-SD (3)	5.8	2.515	13.910	0.782	**0.867**
NMEEF-SD (5)	4.6	2.125	17.295	0.768	0.822
SDIGA-SD (3)	2.8	2.783	8.911	0.653	0.650
SDIGA-SD (5)	4.4	2.525	7.892	0.810	0.630
GAR-SD	7.3	3.439	**39.121**	0.813	0.829

5 Conclusions

In this paper, a new evolutionary multi-objective algorithm for Subgroup Discovery tasks is described. The main feature of this new method is that it can work with both discrete and continuous attributes without previous discretization. The ranges of numeric attributes are obtained in the rules induction process itself. In this way, we ensure that these intervals are the most suitable for maximizing the quality measures.

GAR-SD has been compared with APRIORI-SD, CN2-SD, SDIGA, MESDIF and NMEEF-SD measuring the significance, support and confidence from datasets with different characteristics. This study shows that the algorithm obtains the best results for significance for all the datasets. In addition, good results for support and confidence were obtained. The study evinces that GAR-SD obtains good results for SD tasks. The subgroups obtained by GAR-SD are, therefore, good, useful and representative.

As future work, other measures will be studied, such as coverage, unusualness and accuracy in order to improve the subgroups obtained by the algorithm.

References

1. Bay, S.D., Pazzani, M.J.: Detecting group differences: Mining contrast sets. Data Mining and Knowledge Discovery 5(3), 213–246 (2001)
2. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: Proc. 5th ACM SIGKDD Int. Conf. on KDD and DM, pp. 43–52 (1999)
3. Klösgen, W.: Explora: A multipattern and multistrategy discovery assistant. Advances in Knowledge Discovery and Data Mining, 249–271 (1996)
4. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Proc. of the 1st European Conference on Principles of DM and KDD (PKDD-97), pp. 78–87 (1997)
5. Novak, P.N., Lavrač, N., Webb, G.I.: Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining. Journal of Machine Learning Research 10, 377–403 (2009)
6. Lavrac, N., Kavsek, B., Flach, P.A., Todorovski, L.: Subgroup discovery with CN2-SD. Journal of Machine Learning Research 5, 153–188 (2004)
7. Kavsek, B., Lavrac, N.: APRIORI-SD: Adapting association rule learning to subgroup discovery. Applied Artificial Intelligence 20(7), 543–583 (2006)
8. Atzmüller, M., Puppe, F.: SD-Map - a fast algorithm for exhaustive subgroup discovery. In: Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2006), pp. 6–17 (2006)
9. Klösgen, W., May, M.: Spatial subgroup mining integrated in an object-relational spatial database. In: Proc. 6th European Conf. on Principles and Practice of KDD, pp. 275–286 (2002)
10. Zelezny, F., Lavrac, N.: Propositionalization-based relational subgroup discovery with RSD. Machine Learning 62, 33–63 (2006)
11. Herrera, F.: Genetic Fuzzy Systems: taxonomy, current research trends and propects. Evolutionary Intelligence 1, 27–46 (2008)
12. Del Jesús, M.J., González, P., Herrera, F., Mesonero, M.: Evolutionary fuzzy rule induction process for subgroup discovery: A case study in marketing. IEEE Transactions on Fuzzy Systems 15(4), 578–592 (2007)
13. Berlanga, F., del Jesus, M.J., González, P., Herrera, F., Mesonero, M.: Multiobjective Evolutionary Induction of Subgroup Discovery Fuzzy Rules: A Case Study in Marketing. In: Perner, P. (ed.) ICDM 2006. LNCS (LNAI), vol. 4065, pp. 337–349. Springer, Heidelberg (2006)
14. Carmona, C.J., González, P., del Jesus, M.J., Herrera, F.: NMEEF-SD: Non-dominated Multi-objective Evolutionary algorithm for Extracting Fuzzy rules in Subgroup Discovery. IEEE Transactions on Fuzzy Systems 18(5), 958–970 (2010)
15. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: 13th Int. Joint Conf. Artif. Intell, pp. 1022–1029 (1993)
16. Mata, J., Alvarez, J.L., Riquelme, J.C.: Discovering Numeric Association Rules via Evolutionary Algorithm. In: Proc. of PAKDD, pp. 40–51. Springer, Heidelberg (2002)
17. Goldberg, E.D.: Genetic Algorithms in Search, Optimization and Machine Learning. Wesley, Longman (1989)
18. 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, vol. 13(3), pp. 307–318 (2009)

Gene Regulatory Networks Validation Framework Based in KEGG

Norberto Díaz–Díaz, Francisco Gómez–Vela,
Domingo S. Rodriguez–Baena, and Jesús Aguilar–Ruiz

School of Engineering
Pablo de Olavide University. Spain
{ndiaz,fgomez,dsrodbae,aguilar}@upo.es

Abstract. In the last few years, DNA microarray technology has attained a very important role in biological and biomedical research. It enables analyzing the relations among thousands of genes simultaneously, generating huge amounts of data. The gene regulatory networks represent, in a graph data structure, genes or gene products and the functional relationships between them. These models have been fully used in Bioinformatics because they provide an easy way to understand gene expression regulation.

Nowadays, a lot of gene regulatory network algorithms have been developed as knowledge extraction techniques. A very important task in all these studies is to assure the network models reliability in order to prove that the methods used are precise. This validation process can be carried out by using the inherent information of the input data or by using external biological knowledge. In this last case, these sources of information provide a great opportunity of verifying the biological soundness of the generated networks.

In this work, authors present a gene regulatory network validation framework. The proposed approach consists in identifying the biological knowledge included in the input network. To do that, the biochemical pathways information stored in KEGG database will be used.

1 Introduction

In the last few years, DNA microarray technology has attained a very important role in biological and biomedical research. It enables analyzing the relations among thousands of genes simultaneously. The huge amount of data generated has attracted the attention of a lot of researchers because extracting useful information from it represents a big challenge. For example, inferring gene-gene interactions involved in biological function is a relevant task. To deal with it, different statistical and data mining techniques have been used. Gene Regulatory Networks (GRNs) represent one of the most important approaches to support the identification of regulatory modules. These approaches are able to establish gene-gene interactions and show them in a visual way. There are different models of regulatory networks [6], among which we point out the followings: regulatory

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 279–286, 2011.

networks based on information theory [4,14], logical networks [12,19,3], based on differential equations [6], bayesian networks [7,15] and tree-based networks [16,20]. Regarding the first type, information theory based networks, **correlation networks** represent the simplest architecture in which a non-directed graph with weighted edges according the correlation coefficient between the gene expression levels is used. The **logical networks** were introduced by Kauffman [11] and take into account the different gene states in a binary way. In the case of the **differential equations based networks**, the changes in the gene expression levels are described by means of a differential equation, which is suitable to model complex relationships. Another model are **bayesian networks** that use Bayes rules to show the stochastic behavior of the genetic regulation. And finally, we should point **tree-based networks** that are inspired by model trees as a way to detect linear dependencies between genes and to set a group of gene-gene dependencies.

In the literature, there are a lot of studies about the aforementioned networks models. For example, in the work of Faith et al. [5] the edges are weighted using statistical values from mutual information or Rangel et al. [18] research, where a 39 genes lineal model was inferred about the T-cell activation using temporal gene expression data.

A very important task in all these studies is to assure the network models reliability in order to prove that the used methods are good enough. The most accepted approach is the validation using public data, that is, gene networks are validated comparing them with well-known and accepted data. In this sense, we point out the work of Bickel et al. [2] which is based on ROC curves to carry out the networks validation. In [13], the euclidean distance (L2-Distance) between the final results and the best known result is used as a validation measure.

In conclusion, the GRN validation is a crucial step in the new approaches development process. In this work, a novel validation framework based on the input network biological knowledge identification is proposed. This approach consists of the comparison between the input network and the metabolic processes represented by the biochemical pathways stored in the Kyoto Encyclopedia of Genes and Genomes, KEGG [9,17,10]. Besides, quantitative measures are used in order to estimate the network goodness.

The paper is organized as follows: We start with a description section in which the main features of our approach are described. Following, the methodology is applied to synthetic and real data and the results obtained are fully explained. Finally, the main conclusions are exposed in the section 4.

2 Description

Our approach consists in the comparison between the gene-gene interactions of the input network[1] with the interactions stored in the KEGG database. KEGG project was born in 1995 as a part of the Japanese human genome project.

[1] The input network could be a Gene Regulatory Network or an Influence Gene Network.

KEGG is composed of 16 databases organized in 3 categories: *genetic information*, where, for example, the database KEGG GENES stores information about completely sequenced genomes; *chemical information*, where biochemical reactions are represented in the KEGG REACTION database; and *functional information*, where it is worth to mention the KEGG PATHWAY database, which stores knowledge about different biological process in graph data structures. These graphs are composed by nodes (molecules, proteins or genes) and edges that represents relationships, reactions or interactions between the nodes.

The methodology presented in this work consists in identifying the biological knowledge stored in the input network. To deal with it, a comparison between the input graph and the different pathways stored in KEGG is carried out. For every biochemical pathway it is checked if the metabolic process described by that pathway is contained, totally or partially, in the input network.

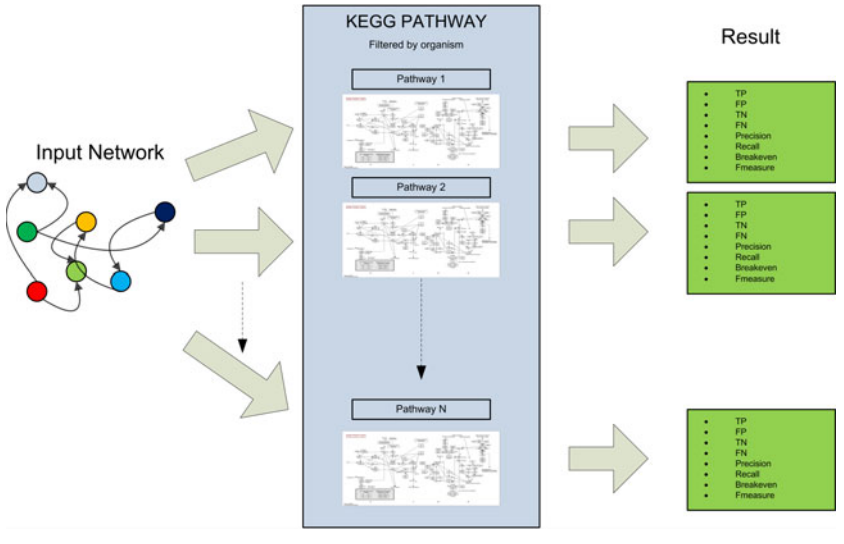


Fig. 1. A schematic comparison methodology representation

A schematic representation of our methodology is depicted in Figure 1. Firstly, the organism o represented in the input network G is detected. In second place, the different pathways, P_o, stored in KEGG for the organism o are extracted. For example, in the figure there are N pathways stored in KEGG for o ($P_o = \{p_1, \ldots, p_N\}$). Finally, N comparisons between the different pathways, $p_i \in P_o$, and the input network G are carried out. Every comparison checks every graph edge against every pathway (p_i) edge, that is, for every edge in G a similar one in p_i will be searched. This comparison will be based on the nodes connected by the edges. It is important to take into account that every node stored in KEGG PATHWAYS can represent one or more genes. So, on the one hand, it would be necessary that at least one of them matches up with a node from the input

network. On the other hand, genes from G not involved in the metabolic process represented by p_i are not took into account to carry out the comparison.

In that way, for every comparison, we will find the differences and similarities between the input network and the information stored in KEGG. Differences are classified as **false positives (FP)**, that is, there is a relation in the input network but not in KEGG, or **false negatives (FN)** if there is a relation in KEGG not detected by the input network. Similarities are classified as **true positives (TP)** or **true negatives (TN)**. If the input network shares a edge between a pair of genes $\{g_i,g_j\}$ with a KEGG pathway, it will classified as a **TP**. We will get a **TN** when a pair of genes $\{g_i,g_j\}$ are not found either in the input network neither in the KEGG patwhay p_i.

Using these definitions, the following measures are calculated:

Coverage. Is the percentage of pathway edges included in the input network. It is calculated by means of the following equation, in which $|p|$ is the number of the edges in pathway p.

$$Coverage = \frac{TP * 100}{|p|}$$

Precision. Shows the relationship between the number of TP and the number of edges in the input network. It is a measure of our network accuracy, represented by values between $[0, 1]$, being 1 the maximum possible accuracy, that is, the 100% of the gene–gene interactions are found in the analyzed pathway.

$$Precision = \frac{TP}{TP + FP}$$

Recall. The recall value shows the relationships between the number of gene–gene interactions in KEGG and in the input network. With values in the following interval, $[0, 1]$, a recall of 1 implies a input network containing all the relationships stored in KEGG.

$$Recall = \frac{TP}{TP + FN}$$

Breakeven. It is the balance between precision and recall. The generated values are in the same interval that the previous ones, being a more sensitive network if the value is equal to 1.

$$Breakeven = \frac{Precision + Recall}{2}$$

F-measure. Is the harmonious measure between precision and recall. Unlike breakeven, that is an arithmetic mean, F-measure is an adjusted mean.

$$F - measure = \frac{2 * Precision * Recall}{Presion + Recall}$$

3 Results

The objective of this section is to prove the usefulness of the gene regulatory network validation framework proposed in this work. To do that, the experimental part is divided in two different phases. Firstly, well-known data will be used in order to demonstrate that the method can detect the quality of the input networks. In second place, the objective will be the opposite, that is, using data without any biological soundness it will be prove that the final results will not be relevant enough.

For the first phase, the network presented by por Uetz et al.[21], composed of 957 relationships between 1004 yeast genes, is used. Applying our novel approach, 95 (total number of sacchyaromyces cerevisiae pathways) comparisons are being carried out using KEGG. It is worth to mention the comparison with pathway ***"Regulation of autophagy – Saccharomyces cerevisiae"*** ($path : sce04140$). Its results are summarized in Table 1. The coverage value, $71,4\%$, shows that the network includes almost three quarters of the gene–gene interactions in the examined pathway. Besides, it shows that there are 10 identical relationships in both networks, 6 in the input network but not in the pathway and 9 interactions in the pathway that are not found in the input network. Finally, the rest of the measures described in section 2 are presented.

Table 1. Pathway *sce*04140 comparison results

	Pathway sce:04140
Coverage	71,4%
TP	10
FP	6
FN	9
Precision	0,625
Recall	0,526
Breakeven	0,575
F-measure	0,571

The values of Precision and Coverage stored in Table 1 show that the biological functionality of the pathway ***(sce:04140)*** is contained by the input GRN. Thus, we can say that the input network is reliable based on current knowledge because it contains relevant biological information.

In the second phase, the behavior of our methodology using non-relevant biological information is checked. To deal with it, we modify the same network used in the first part adding random relationships. That is, in a loop process composed by 10 iterations, the number of random relationships added to the network are increased a 10% every iteration. Besides, in every iteration, 100 artificial networks are generated and compared with pathway $path : sce04140$. In that way, 1000 different networks will be validated and their results are summarized in Figure 2.

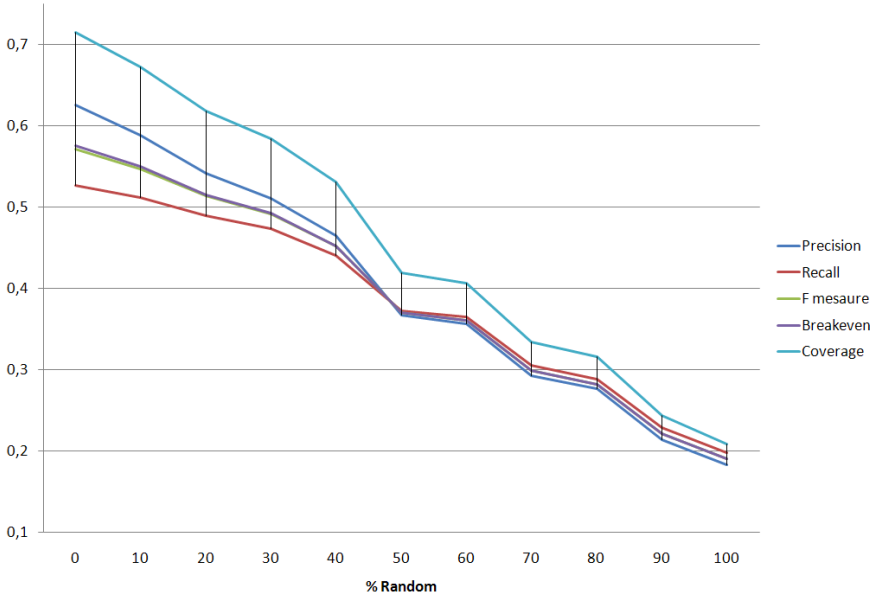


Fig. 2. Evolution of the proposed measures regarding pathway *sce* : 04140

Figure 2 shows, for every iteration, the arithmetic mean of the values obtained by the measures described in section 2. It can be observed that the different measures follow a similar behavior, like in the extreme case of measures *Breakheaven* and *F–measure*. The *coverage* measure starts with a value up to 70% when the random level is null and ends with values near to 20% when the random level reaches its biggest value. This is the more emphasized decrease in the figure, being the expected behavior in the sense that this measure provides better results when the input network shares a big number of edges with the analyzed pathway. The same occurs whith *precision*. However, *recall* measure shows the lightest variation in all the graphic, starting with values near to 0.5 and ending with values near to 0.2.

In conclusion, these results prove that our methodology distinguishes the differences between a gene regulation network with biological relevance and an input network without any biological soundness. However, this experiments make clear that the information stored in KEGG is not suitable in some cases, generating non-real differences between the analyzed pathway and the input network.

4 Conclusions

A first approach of a genes network validation framework is proposed. The method presented, based on the metabolic process information stored in KEGG, consists of identifying known biological knowledge included in the input network.

One of the main features is represented by the different measures proposed to check the goodness of a certain network.

The usefulness of the methodology has been checked using biological well-known data, obtained from literature, as well as artificial data. To be precise, a gene network published by Uetz et al. as real data and, a random modification of it has been used as artificial data. The final results shows that the our methodology is able to measure every kind of result appropriately.

However, it seems that the information stored in KEGG is insufficient in some cases. Because of this, and as a future work, we will incorporate additional information stored in other databases, like Reactome [8] o Enzyme [1] in order to generate more complete and accurate measures. Finally, we will provide different levels of accuracy to the mesures in order to take into account the indirect relationships between genes.

References

1. Bairoch, A.: The enzyme database in 2000. Nucl. Acids Res. 28(1), 304–305 (2000)
2. Bickel, D.R., Montazeri, Z., Hsieh, P.-C., Beatty, M., Lawit, S.J., Bat, N.J.: Gene network reconstruction from transcriptional dynamics under kinetic model uncertainty: a case for the second derivative. Bioinformatics 25(6), 772–779 (2009)
3. Bornholdt, S.: Boolean network models of cellular regulation: prospects and limitations. Journal of the Royal Society Interface 5, S85–S94 (2008)
4. Butte, A.J., Kohane, I.S.: Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In: Pacific Symposium on Biocomputing, pp. 418–429 (2000)
5. Faith, J.J., Hayete, B., Thaden, J.T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J.J., Gardner, T.S.: Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biol. 5(1), 8 (2007)
6. Hecker, M., Lambeck, S., Toepfer, S., Someren, E.v., Guthke, R.: Gene regulatory network inference: Data integration in dynamic models a review. Biosystems 96(1), 86–103 (2009)
7. Heckerman, D.: A tutorial on learning with bayesian networks. Technical report, Microsoft Research, MSR–TR- 95–06 (1995)
8. Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., Jassal, B., Gopinath, G.R., Wu, G.R., Matthews, L., Lewis, S., Birney, E., Stein, L.: Reactome: a knowledgebase of biological pathways. Nucleic acids research 33(Database issue), D428–D432 (2005)
9. Kanehisa, M., Goto, S.: Kegg: Kyoto encyclopedia of genes and genomes. Nucl. Acids Res. 28(1), 27–30 (2000)
10. Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., Hirakawa, M.: Kegg for representation and analysis of molecular networks involving diseases and drugs. Nucleic acids research 38(Database issue), D355–D360 (2010)
11. Kauman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology 22, 437–467 (1969)
12. Kauman, S.A., Glass, K.: The logical analysis of continuous, nonlinear biochemical control networks. Journal of Theoretical Biology 39, 103–129 (1973)

13. Lippert, C., Ghahramani, Z., Borgwardt, K.M.: Gene function prediction from synthetic lethality networks via ranking on demand. Bioinformatics 26(7), 912–918 (2010)
14. Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R.D., Califano, A.: Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC bioinformatics 7 (Suppl. 1) (2006)
15. Needham, C.J., Bradford, J.R., Bulpitt, A.J., Westhead, D.R.: A primer on learning in bayesian networks for computational biology. PLoS Comput. Biol. 3(8), 129 (2007)
16. Nepomuceno-Chamorro, I.A., Aguilar-Ruiz, J.S., Riquelme, J.S.: Inferring gene regression networks with model trees. BMC Bioinformatics 11, 517–528 (2010)
17. Okuda, S., Yamada, T., Hamajima, M., Itoh, M., Katayama, T., Bork, P., Goto, S., Kanehisa, M.: Kegg atlas mapping for global analysis of metabolic pathways. Nucleic acids research 36(Web Server issue), gkn282+ (2008)
18. Rangel, C., Angus, J., Ghahramani, Z., Lioumi, M., Sotheran, E., Gaiba, A., Wild, D.L., Falciani, F.: Modeling t-cell activation using gene expression profiling and state-space models. Bioinformatics 20(9), 1361–1372 (2004)
19. Shmulevich, I., Dougherty, R., Kim, S., Zhang, W.: Probabilistic boolean networks: A rule-based uncertainty model for gene regulatory networks. Bioinformatics 18(2), 261–274 (2002)
20. Soinov, L.A., Krestyaninova, M.A., Brazma, A.: Towards reconstruction of gene networks from expression data by supervised learning. Genome Biology 4, 6 (2003)
21. Uetz, P., Giot, L., Cagney, G., Mansfield, T.A., Judson, R.S., Knight, J.R., Lockshon, T., Narayan, V., Srinivasan, M., Pochart, P., Qureshi-Emili, A., Li, Y., Godwin, P., Conover, D., Kalbfleisch, P., Vijayadamodar, G., Yang, M., Johnston, M., Fields, S., Rothberg, J.M.: A comprehensive analysis of protein protein interactions in saccharomyces cerevisiae. Nature 403(6770), 623–627 (2000)

Computational Intelligence Techniques for Predicting Earthquakes

F. Martínez-Álvarez[1], A. Troncoso[1], A. Morales-Esteban[2], and J.C. Riquelme[3]

[1] Department of Computer Science, Pablo de Olavide University of Seville, Spain
{fmaralv,ali}@upo.es
[2] Department of Continuum Mechanics, University of Seville, Spain
ame@us.es
[3] Department of Computer Science, University of Seville, Spain
riquelme@us.es

Abstract. Nowadays, much effort is being devoted to develop techniques that forecast natural disasters in order to take precautionary measures. In this paper, the extraction of quantitative association rules and regression techniques are used to discover patterns which model the behavior of seismic temporal data to help in earthquakes prediction. Thus, a simple method based on the k–smallest and k–greatest values is introduced for mining rules that attempt at explaining the conditions under which an earthquake may happen. On the other hand patterns are discovered by using a tree-based piecewise linear model. Results from seismic temporal data provided by the Spanish's Geographical Institute are presented and discussed, showing a remarkable performance and the significance of the obtained results.

Keywords: time series, quantitative association rules, regression.

1 Introduction

A time series is a sequence of values observed over time and, therefore, chronologically ordered. Given this definition, it is usual to find data that can be represented as time series in many research fields.

The study of the past behavior of a variable may be extremely valuable to predict its future behavior. Assuming that the nature of the earthquakes time series is stochastic, clustering techniques have shown that these time series exhibit some temporal patterns, making the modeling and subsequent prediction possible [11].

This paper analyzes and forecasts earthquakes time series by means of the application of two classical techniques: Quantitative association rules (QAR) and regression.

A revision of the latest published works reveals that the amount of metaheuristics and search algorithms related to association rules with continuous attributes is limited. Nevertheless, a classifier was presented in [13] to extract quantitative association rules from unlabeled data streams. The main novelty

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 287–294, 2011.

of this approach lied on its adaptability to on-line gathered data. Also, a metaheuristic based on rough particle swarm techniques was presented in [1]. In this case, the special feature was the obtention of the values determining the intervals of the association rules. They also evaluated and tested several new operators in synthetic data. A multi-objective pareto-based genetic algorithm was presented in [2]. The fitness function was formed by four different objectives: support, confidence, comprehensibility of the rule (aimed at being maximized) and the amplitude of the intervals that forms the rule (intended to be minimized). The work published in [17] presented a new approach based on three novel algorithms: Value-interval clustering, interval-interval clustering and matrix-interval clustering. Their application was found especially useful when mining complex information. Another genetic algorithm was used in [16] in order to obtain numeric association rules. However, the unique objective to be optimized in the fitness function was the confidence. To fulfill this goal, the authors avoided the specification of the actual minimum support, which is the main contribution of this work. Finally, an extension of the well-known binary-coded CHC algorithm is presented in [10] for finding existing relations between atmospheric pollution and climatological conditions.

Regression techniques have been widely used for forecasting time series [5]. Thus, an empirical study on sea water quality prediction can be found in [7]. Hatzikos et al. faced the problem of forecasting water quality based on underwater sensors measurements, by means of a large variety of both linear and nonlinear methods. Also, a new methodology to build regression trees was introduced in [3]. The authors transformed quantitative data into statistical moments, and constructed a tree to estimate the forecasting interval of the target variable. Last, the problem of predicting the machinery degradation and trending of fault propagation before reaching the alarm was studied in [12]. In particular, the authors proposed an approach based on regression trees to forecast such time series.

The rest of the paper is divided as follows. Section 2 provides the methodology used in this work. The results of the approach are reported in Section 3. Finally, Section 4 discusses the achieved conclusions.

2 Methodology

The methods used to extract knowledge from earthquakes time series are described in this section. The goal is to find patterns in data that precede the appearance of earthquakes with a given magnitude.

2.1 Association Rules Mining

Let $F = \{F_1, ..., F_n\}$ be a set of features with values in $\mathbb{R}$ describing an earthquake. The desired rules are defined by the following equation:

$$\bigwedge_{i=1,\ldots,n-1} F_i \in [l_i, u_i] \Rightarrow F_n \in [l_n, u_n] \quad (1)$$

where l_i and u_i represents the lower and upper limits of the interval for F_i, respectively and the limits l_n and u_n are given depending on the objective of the problem to be solved. In the context of seismic time series, F_n represents the earthquake magnitude to be predicted and the limits l_n and u_n depend on the required size of the earthquakes to be forecasted.

The proposed method to obtain QAR is described as follows. First, the dataset is sorted by the feature F_n, that is, by the consequent of the rule. Once the limits $[l_n, u_n]$ are set, the range of the remaining features F_i is calculated as:

$$R(F_i) = \{F_i \text{ such that } F_n \in [l_n, u_n]\} \qquad i = 1, ..., n-1 \tag{2}$$

Let f_i^M and f_i^m with $i = 1, ..., n-1$ two functions defined by:

$$\begin{array}{rl} f_i^M : \{1, ..., \#(R(F_i))\} & \longrightarrow R(F_i) \\ k & \longrightarrow f_i^M(k) = k \text{ greatest value of } R(F_i) \end{array} \tag{3}$$

$$\begin{array}{rl} f_i^m : \{1, ..., \#(R(F_i))\} & \longrightarrow R(F_i) \\ k & \longrightarrow f_i^M(k) = k \text{ smallest value of } R(F_i) \end{array} \tag{4}$$

where $\#(R(F_i))$ is the number of elements of the set $R(F_i)$.

Let S_i be the set of pair of values such that the amplitude of the interval to be searched for the feature F_i is sufficiently small. That is,

$$S_i = \{(k_1, k_2) \text{ such that } f_i^M(k_2) - f_i^m(k_1) \leq MAX_i\} \tag{5}$$

where MAX_i is the maximum allowed amplitude for the feature F_i which is a given parameter depending on the desired rules.

Thus, for any value $(k_1^i, k_2^i) \in S_i$, the rules built by the k-greatest and k-smallest values are:

$$\bigwedge_{i=1,...,n-1} F_i \in [f_i^m(k_1^i), f_i^M(k_2^i)] \Rightarrow F_n \in [l_n, u_n] \tag{6}$$

2.2 Regression: M5P Algorithm

The second method used to obtain patterns in seismic time series is the M5P algorithm available in WEKA [4]. The M5P approach [15] extends to the M5 algorithm by adding missing values techniques and transformation of features from discrete values to binary values. The algorithm M5 [14] provides a conventional decision-tree with linear regression functions at the nodes. The tree is obtained by a classical induction algorithm but the splits are obtained by maximizing the reduction of the variance and not maximizing the gain of information.

Once the tree has been built, the method computes a linear model for each node. Later the leaves of the tree are pruned while the error decreases. For each node, the error is the mean of the absolute value of the difference between the

predicted and actual values for each example reaching such node. This error is weighted depending on the number of examples which reach that node. The process is repeated until all examples are covered for one o more rules.

Thus, M5P generates models that are compact and relatively comprehensible.

3 Results

This section presents the results obtained from the application of the approaches introduced in Section 2. In particular, Section 3.1 provides a description of the data used. Sections 3.2 and 3.3 gather all relevant results mined by means of association rules and decision-tree techniques, respectively.

3.1 Data Description

The dataset used in this work has been retrieved from the catalogue of Spanish's Geographical Institute (SGI), which contains the location and magnitude of Spanish earthquakes.

Additionally, the b–value parameter of the Gutenberg–Richter law has been calculated, as it reflects the tectonics and geophysical properties of the rocks as well as the fluid pressure variations in the characterized surface [9].

Thus, each sample forming the dataset is composed by four attributes: Current earthquake magnitude, time when the earthquake occurred, associated b-value, and magnitude of the previously occurred earthquake. Note that earthquakes with magnitude lower than 3.0 have been removed from the dataset, and both aftershocks and foreshocks have been removed to avoid dependent data, as recommended in [8].

Despite the Iberian Peninsula is divided in 27 seismogenic areas according to SGI, only areas 26 and 27 (Alboran Sea and Western Azores–Gibraltar Fault, respectively) have been studied, since they are the most active ones [11]. The considered earthquakes date from 1981 to 2008, having been analyzed a total of 873 quakes.

3.2 Quantitative Association Rules Extraction

All mined association rules to forecast earthquakes are now introduced and discussed. As the goal is to find patterns that precede quake occurrences, the magnitude of the current earthquake, M_c, has been forced to be the only attribute in the consequent.

The M_c attribute has been divided in three non-overlapped intervals: [3.0, 3.5) or small earthquakes, [3.5, 4.4) or medium earthquakes, and [4.4, 6.2] or large earthquakes (note that the largest retrieved earthquake magnitude is 6.2).

Tables 1, 2, and 3 show the rules extracted for large, medium and small earthquakes, respectively. Note that Δb and Δt represent the increment of the b–value and the time elapsed between the previous and current earthquake, respectively. Also, the magnitude of the earthquake occurred prior the current one, M_p, has been

Table 1. Association rules with consequent $M_c \in [4.4, 6.2]$

Id	Antecedent	Conf. (%)	Sup. (%)	Lift
#1	$\Delta t \in [0.02, 0.08] \wedge \Delta b \in [-0.16, -0.10] \wedge M_p \in [3.0, 3.4]$	75.0	5.7	12.4
#2	$\Delta t \in [0.00, 0.07] \wedge \Delta b \in [-0.12, -0.05] \wedge M_p \in [3.5, 4.9]$	87.5	13.2	14.4
#3	$\Delta t \in [0.00, 0.33] \wedge \Delta b \in [-0.11, -0.01] \wedge M_p \in [5.0, 6.2]$	80.0	7.6	13.2

divided in non-overlapped intervals, and therefore, transactions forming the data can be covered only by one rule. Finally, all rules have been assessed by means of three well-known and widely used indices: Confidence, support, and lift [6].

The best rules mined for large earthquakes ($M_c \in [4.4, 6.2]$) are shown in Table 1. These rules share a common feature, which is that they all present remarkable and negative Δb. Moreover, Δt is small in all rules, except for rule #3, which allows time intervals up to 0.33. From the 53 earthquakes that satisfy that $M_c \in [4.4, 6.2]$, 14 are covered by rules #1, #2 and #3, which represents a support of 26.4%. On the other hand, it is noticeable the high confidence reached by all of them: 80.8% on average. Finally, the interestingness of the rules (or lift) is 13.3 on average. Assuming that a lift greater than 1 leads to consider the rule as interesting [6], the obtained values indicate that the extracted rules provide meaningful knowledge.

Table 2 shows the QAR obtained for medium earthquakes, that is, with $M_c \in [3.5, 4.4)$. The most significative feature that share all the rules is that the b–value does not vary much (its value ranges from $\Delta b = -0.07$ to $\Delta b = 0.02$). Also remarkable is that the occurrence of these earthquakes takes place after moderately short time periods (the time elapsed between earthquakes varies from $\Delta t = 0.00$ to $\Delta t = 0.20$). As for the quality of the results, 86 earthquakes out of 344 were covered by rules #4, #5 and #6, which means a support of 25.0%. The confidence was of 76.0% on average which can be considered high. Last, the lift measure also confirms that the rules are high quality, since it has values greater than 1, in particular, 1.9 on average.

Table 3 represents the best QAR discovered for small earthquakes ($M_c \in [3.0, 3.5)$). The b–value is now characterized by moderate and positive increments (Δb ranges from 0.01 to 0.04). Moreover, in contrast to what happens with medium and large earthquakes, the time elapsed is high, varying from $\Delta t = 0.10$ to $\Delta t = 0.32$. A total of 476 small earthquakes were retrieved, from which 46 have been covered by rules #7, #8 and #9, which imply a support of 9.7%. Especially noticeable is the confidence reached by these rules which is 85.7% on average. Again, the lift measure is greater than 1 for all rules, in particular, 1.7 on average.

Table 2. Association rules with consequent $M_c \in [3.5, 4.4)$

Id	Antecedent	Conf. (%)	Sup. (%)	Lift
#4	$\Delta t \in [0.04, 0.20] \wedge \Delta b \in [-0.07, -0.01] \wedge M_p \in [3.0, 3.5]$	79.0	8.7	2.0
#5	$\Delta t \in [0.00, 0.02] \wedge \Delta b \in [-0.01, 0.00] \wedge M_p \in [3.6, 4.5]$	78.6	12.8	2.0
#6	$\Delta t \in [0.00, 0.05] \wedge \Delta b \in [-0.02, 0.02] \wedge M_p \in [4.6, 5.9]$	70.6	3.6	1.8

Table 3. Association rules with consequent $M_c \in [3.0, 3.5)$

Id	Antecedent	Conf. (%)	Sup. (%)	Lift
#7	$\Delta t \in [0.13, 0.32] \wedge \Delta b \in [0.01, 0.04] \wedge M_p \in [3.0, 3.2]$	100	2.5	1.8
#8	$\Delta t \in [0.10, 0.19] \wedge \Delta b \in [0.01, 0.03] \wedge M_p \in [3.3, 3.4]$	88.0	4.6	1.6
#9	$\Delta t \in [0.11, 0.32] \wedge \Delta b \in [0.00, 0.03] \wedge M_p \in [3.5, 5.7]$	85.7	2.5	1.6

3.3 M5P Results

This section provides the result obtained from the application of the M5P regressor. Fig. 1 illustrates the tree built by this algorithm. Thus, M5P found four linear models (LM), whose equations are listed below:

$$\text{LM 1: } M_c = -0.0160\Delta t - 11.2781\Delta b + 0.3237 M_p + 2.3766 \tag{7}$$

$$\text{LM 2: } M_c = -0.0795\Delta t - 0.4022\Delta b + 0.1889 M_p + 2.7826 \tag{8}$$

$$\text{LM 3: } M_c = -0.0955\Delta t - 0.4022\Delta b + 0.0213 M_p + 3.2495 \tag{9}$$

$$\text{LM 4: } M_c = -0.3696\Delta t - 0.4206\Delta b + 0.0096 M_p + 3.2060 \tag{10}$$

The analysis of this model reveals that the b–value is the most significative attribute, as it appears in the two first levels of the tree. Also, the coefficients corresponding to b–value have the greatest weights in the linear models.

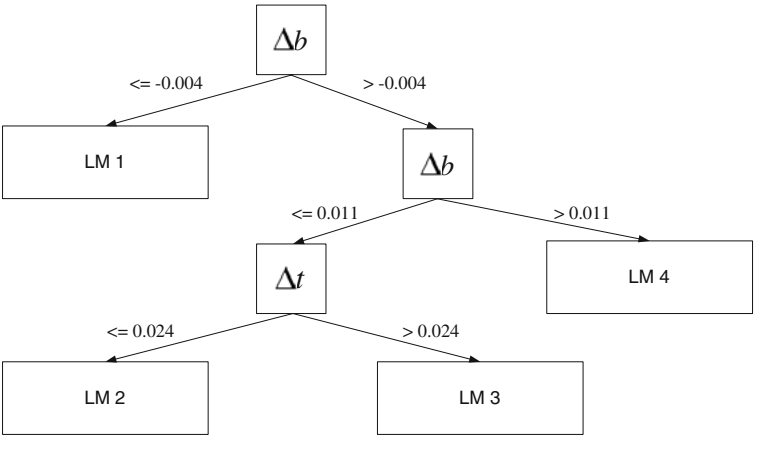


Fig. 1. Tree built with M5P algorithm

The first cutoff is set for $\Delta b = -0.004$. Thus, the first linear model, LM 1, is found when $\Delta b \leq -0.004$. This model has the biggest absolute value for the

Δb coefficient (a value of -11.2781). Moreover, as this coefficient is negative, it can be stated that the smaller is the value of Δb, the bigger is the earthquake magnitude. On the other hand, the coefficient of M_p is positive (a value of 0.3237), which leads to conclude that M_c is directly related to M_p. In other words, the magnitude of the current earthquake has a direct relation with the magnitude of the previous one.

The earthquakes occurred with $\Delta b > -0.004$ are modeled by three linear models (LM 2, LM 3 and LM 4). All of them present similar Δb coefficients, which involves inverse relation with the magnitude of the current earthquake, that is, the bigger is Δb, the smaller is M_c. Nevertheless, its influence is more moderate than that of LM 1.

The second cutoff is set for $\Delta b = 0.011$. Thus, when $\Delta b > 0.011$ the LM 4 model is provided (see equation (10)). In this model, the most significative coefficient is that corresponding to Δt with a weight of -0.3696, revealing that the longer is the time elapsed, the smaller is the magnitude of the current earthquake. It is also notable that the magnitude of the previous earthquake does not influence much in this model as it is weighted by 0.0096.

When the b–value varies between -0.004 and 0.011, the model proposes two different linear models (LM 2 and LM 3), depending on the time elapsed between the previous and current earthquake. Although both linear models are quite similar, when the time elapsed is less or equal than 0.024 (LM 2 model), the magnitude of the previous earthquake influences much more than when it is greater than 0.024 (LM 3 model) as the coefficient of M_p is 0.1889 in LM 2 versus 0.0213 in LM 3.

Finally, a measure of the quality of results is now discussed. The tree presents a correlation coefficient of 0.67. The mean absolute error is 0.26 and the root mean squared error is 0.35. These errors are considered satisfactory given the stochastic nature of the problem studied.

4 Conclusions

Earthquake data from two particular areas of the Iberian Peninsula have been successfully mined by means of two different techniques: QAR and the M5P algorithm. In particular, QAR with a confidence of 83.0% and a lift of 5.6 on average have been discovered and a regression-tree with an error of 0.35 has been built. Both techniques have discovered the great influence that the b–value has in earthquakes occurrences as its variation along with the time elapsed have shown to be useful to model different earthquakes. Thus, the patterns discovered before an earthquake takes place may be useful in subsequent predictions.

Acknowledgments

The financial support from the Spanish Ministry of Science and Technology, project TIN2007-68084-C-02, and from the Junta de Andalucía, project P07-TIC-02611, is acknowledged.

References

1. Alatas, B., Akin, E.: Rough particle swarm optimization and its applications in data mining. Soft Computing 12(12), 1205–1218 (2008)
2. Alatas, B., Akin, E., Karci, A.: MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules. Applied Soft Computing 8(1), 646–656 (2008)
3. Alberg, D., Last, M., Ben-Yair, A.: Induction of mean output prediction trees from continuous temporal meteorological data. Journal of Applied Quantitative methods 4(4), 485–494 (2009)
4. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Data Mining. In: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco (1999)
5. Fu, T.-C.: A review on time series data mining. Engineering Applications of Artificial Intelligence 24(1), 164–181 (2010)
6. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Computing Surveys 38(3), Article 9 (2006)
7. Hatzikos, E.V., Tsoumakas, G., Tzanis, G., Bassiliades, N., Vlahavas, I.I.: An empirical study on sea water quality prediction. Knowledge-Based Systems 21, 471–478 (2008)
8. Kulhanek, O.: Seminar on b–value. Technical report, Department of Geophysics, Charles University, Prague (2005)
9. K. Lee and W. S. Yang. Historical seismicity of Korea. Bulletin of the Seismological Society of America, 71(3):846–855, 2006.
10. Martínez-Ballesteros, M., Troncoso, A., Martínez-Álvarez, F., Riquelme, J.C.: Mining quantitative association rules based on evolutionary computation and its application to atmospheric pollution. Integrated Computer-Aided Engineering 17(3), 227–242 (2010)
11. Martínez-Álvarez, F., Morales-Esteban, A., Troncoso, A., de Justo, J.L., Rubio-Escudero, C.: Pattern recognition to forecast seismic time series. Expert Systems with Applications 37(12), 8333–8342 (2010)
12. Oh, M.S., Tan, A., Tran, V., Yang, B.S.: Machine condition prognosis based on regression trees and one-step-ahead prediction. Mechanical Systems and Signal Processing 22, 1179–1193 (2008)
13. Orriols-Puig, A., Casillas, J., Bernadó-Mansilla, E.: First approach toward online evolution of association rules with learning classifier systems. In: Proceedings of the Computation Conference Genetic and Evolutionary GECCO 2008, pp. 2031–2038 (2008)
14. Quinlan, J.R.: Learning with continous classes. In: Proceedings of the 5th Australian Joint Conference on Artificial Intelligence, pp. 343–348. World Scientific, Singapore (1992)
15. Wang, Y., Witten, I.H.: Induction of model trees for predicting continous classes. In: Proocedings of the poster papers of the European Conference on Machine Learning, pp. 128–137 (1997)
16. Yan, X., Zhang, C., Zhang, S.: Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. Expert Systems with Applications 36(2), 3066–3076 (2009)
17. Yin, Y., Zhong, Z., Wang, Y.: Mining quantitative association rules by interval clustering. Journal of Computational Information Systems 4(2), 609–616 (2008)

Reduct-Based Analysis of Decision Algorithms: Application in Computational Stylistics

Urszula Stańczyk

Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland

Abstract. Computational stylistics focuses on description and quantifiable expression of linguistic styles of written documents that enables author characterisation, comparison, and attribution. It is a case when observation of subtle relationships in data sets is required, with domain knowledge uncertain. Therefore, techniques from the artificial intelligence area, such as Dominance-based Rough Set Approach (DRSA), are well suited to handle the problem. DRSA enables construction of a rule-based classifier consisting of decision rules, selection of which can greatly influence classification accuracy. The paper presents research on application of DRSA classifier in author recognition for literary texts, with considerations on the classifier performance based on an analysis of relative reducts, such subsets of features that maintain classification properties.

Keywords: DRSA, Classifier, Computational Stylistics, Reduct, Decision Algorithm, Feature Selection.

1 Introduction

Computational stylistics or stylometry yields observations on writing styles of authors, expressed in terms of quantifiable measures. These measures can be exploited for characterisation of writers, finding some similarities and differentiating features, as well as for authorship attribution [5].

Author characterisation can be considered as the most basic of three stylometric tasks, as it is the assumption that enables the other two. Even though the authors use language features to some degree subconsciously, still their individual styles can be detected by means of so-called *authorial invariant*, a set of markers that remain the same for texts by one author and distinctively different for texts written by others. That is the fundamental concept of stylometry.

The choice of characteristic features describing texts within the analysis is one of crucial decisions to be made, as there is no consensus within the stylometric research community [1] as to which textual markers should be employed. This is a case where domain knowledge is insufficient to make an informed selection of characteristic features for a classifier, which, however, is essential to its correct recognition ratio. It could be argued that the best approach is to gather as much data as possible and leave it to the methodology applied to discard what is

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 295–302, 2011.

irrelevant or excessive as techniques exploited in such processing possess their own inherent mechanisms of dimensionality reduction.

In cases of processing data that is incomplete and uncertain, techniques from the artificial intelligence area have often proven their efficient performance. Classification with their help can be attempted with a rule-based or connectionist approach [8]. Rough set theory is an example of the first type and within the methodology reduction of unnecessary data is typically obtained in two ways. Firstly, by finding relative reducts - such subsets of conditional attributes that maintain classification properties of a decision table, and secondly, by discarding some of the calculated decision rules, basing on their support and length.

In the past research [9] it has been shown that without undermining the power of the classifier some attributes can be disregarded basing on their numbers of occurrences in construction of relative reducts and rules. This paper presents results of the research on performance of DRSA classifiers applied within the stylometric task of authorship attribution, with selection of rules to construct a decision algorithm based on cardinalities of the calculated relative reducts.

2 Computational Stylistics

Computational stylistics or stylometry relies on individuality of writing styles expressed by such a combination of usage for lexical, syntactic, structural, or content-specific markers that is unique for a writer [2]. This fundamental concept that a style can be defined and expressed in quantitative terms allows for the discovery of characteristics of authors, style comparison, proving or disproving authorship. Among possible applications there could be mentioned establishing authenticity of historic documents, or plagiarism detection.

As some striking features of texts are easily detectable for anyone who would like to imitate someone else's style, they are unreliable as descriptors. Instead, there are often used frequencies of usage of most common words and punctuation marks. These lexical and syntactic markers are used on subconscious level and by that much harder to imitate. Still, for the whole textual analysis to be reliable, the corpus of texts for processing has to be sufficiently wide [5].

To satisfy these base requirements as the input data in the experiments there were chosen works of two writers from XIXth century, Henry James and Thomas Hardy. The samples were created by computing characteristics for markers within parts taken from novels. These fragments were of approximately the same length, typically corresponding to chapters. For the learning set there were 180 samples, and for two testing sets 80 samples (Test80) and 48 samples (Test48) respectively. The second testing set was used in the final phase of experiments for verification of findings, therefore unless otherwise specified, the use of Test80 is assumed.

To allow for author attribution the base set of descriptors was built with frequencies of usage for the arbitrarily selected common function words and punctuation marks (25 attributes) as follows: but, and, not, in, with, on, at, of, this, as, that, what, from, by, for, to, if, a fullstop, a comma, a question mark, an exclamation mark, a semicolon, a colon, a bracket, a hyphen.

3 Data Mining with DRSA

Classical Rough Set-based Approach (CRSA), invented by Z. Pawlak [4], perceives the universe in terms of objects that can or cannot be discerned basing on values of describing them conditional attributes. The indiscernibility relation enables classification only in nominal sense, which means processing abstract or discrete data. To allow for real-valued data sets there can be used discretisation, or there can be employed Dominance-based Rough Set Approach (DRSA).

In DRSA the Pareto or *dominance* principle replaces the indiscernibility, stating that if x is at least as good as y with respect to conditional attributes, then x should be classified at least as good as y. Such reasoning observes ordering in value sets and allows for not only nominal but also ordinal classification [7].

The first step within processing is the construction of the decision table, containing the whole knowledge about the universe. The columns of the table correspond to attributes, while rows specify values of these attributes for objects of the universe. Such table quite often contains excessive data, hence after checking the consistency of the table, the dimensionality reduction is attempted.

To reduce dimensionality, rough set methodology offers two mechanisms. The first one determines relative reducts that are such subsets of conditional attributes that keep intact the classification properties of the decision table [6]. With the help of reducts there are constructed decision algorithms consisting of rules specifying conditions to be met for any decision to be applicable. When building decision algorithms there can be generated rules just providing a minimal cover of learning samples (61 rules) or all rules on examples (46,191 rules).

Quite often a number of relative reducts is high, making a selection of one for further processing problematic, especially with insufficient domain knowledge about the influence of particular attributes on author attribution process. When minimal cover decision algorithm gives unsatisfactory results (only 45% correct decisions) and all rules on examples algorithm contains far too many rules (as in the studied case) to be efficiently employed, a question how to construct an optimised classifier arises. It can be done by selecting the rules with some required length and support. It can also be performed by selecting the rules with attributes meeting some additional criteria, independent on domain knowledge.

Past research [9] shows that as the criteria for attributes there can be used information on how often they are exploited in reducts and decision rules. Such reasoning means assuming that all reducts and all rules are of the same merit. Yet reducts have different cardinalities, while rules vary in length and support. These considerations have led to motivation for the presented research, where attributes are studied through more detailed analysis of reducts they belong to.

4 Reduct-Based Analysis of Attributes

The past research concerned how often condition attributes appear in reducts and rules [9]. Such approach assumes that all reducts are of the same merit. From DRSA point of view they in fact are: they all ensure keeping the classification

ratio. Yet the reducts vary in cardinality. So, if an attribute appears several times in shorter reducts is it the same when it is used in longer ones?

A short relative reduct means less processing and that in the learning set there is present some specific pattern, allowing for classification with fewer attributes. Yet we cannot be sure that this pattern is also present in the testing set. If it is then the classification is ideal 100% recognition, but that is rarely the case. Patterns found in the learning samples are typically present in the testing set only to some degree. Hence for all conditional attributes there is performed analysis with respect to cardinalities of reducts built with them (Table 1). In the experiments there were 6,664 reducts found, with lengths varying from 4 to 14.

Table 1. Analysis of attributes based on relative reducts

Red. card.	4	5	6	7	8	9	10	11	12	13	14			
Nr of red.	6	160	635	1323	1210	1220	1245	582	233	46	4	Quality indicators		
Attr.	Nr of reducts of specific card. for attributes											QI_1	QI_2	QI_3
;	0	3	87	318	517	644	658	326	157	28	2	2740	307.52	25137
,	0	5	70	308	423	707	811	409	174	32	4	2943	323.62	27517
with	0	6	102	355	337	382	482	299	157	39	2	2161	245.09	19789
for	0	8	74	247	333	276	364	195	79	29	4	1609	184.73	14527
but	0	9	33	121	161	144	223	132	57	10	3	893	100.74	8212
(	0	10	44	152	179	261	329	218	158	41	3	1395	151.67	13318
what	0	11	124	203	204	226	355	176	95	19	2	1415	163.49	12787
at	0	12	118	429	477	466	606	302	146	28	1	2585	297.20	23293
as	0	14	137	355	288	366	498	278	141	28	3	2108	242.21	19111
that	0	14	69	160	180	274	337	189	93	23	4	1343	150.78	12430
-	0	14	93	316	405	372	415	241	144	32	3	2035	233.48	18415
this	0	15	117	348	457	549	555	331	169	41	3	2585	293.38	23604
of	0	16	177	580	625	771	775	409	103	19	3	3478	404.28	30915
to	0	18	103	427	446	466	605	279	125	27	1	2497	287.72	22443
?	0	26	124	212	260	275	379	283	121	28	4	1712	195.35	15688
!	0	39	183	396	489	494	483	221	52	9	2	2368	284.44	20453
on	1	9	184	400	488	631	808	364	157	38	3	3083	351.08	28040
:	1	23	130	351	342	522	602	265	116	29	3	2384	273.81	21536
if	1	29	138	265	141	247	344	299	96	22	2	1584	183.39	14378
in	2	50	165	434	374	519	742	305	120	15	0	2726	317.49	24359
.	3	74	226	477	560	696	729	300	108	16	1	3190	379.16	27929
from	3	90	186	398	421	460	337	207	132	36	3	2273	276.84	19613
by	3	108	228	461	581	598	479	138	47	5	0	2648	330.02	22114
not	4	53	332	579	494	518	512	233	49	4	0	2778	345.72	23263
and	6	144	566	969	498	116	22	3	0	0	0	2324	340.67	16204

The quality indicators in the three right-most columns are calculated as follows. QI_1 reflects the number of occurrences in reducts regardless of their cardinality. QI_2 is calculated as the sum of numbers of occurrences in reducts of specific length divided by these lengths. QI_3 is a sum of products: reduct length multiplied by the number of occurrences in such reducts for all attributes.

5 Obtained Results

Sorted values of the quality indicators, calculated during the analysis of reduct cardinalities, led to the ordering of attributes given in Table 2. Order 1 reflects the ordering used in the past research [9]. Order 4 takes into account only the shortest reducts (with cardinality 4 or 5) for all attributes and their numbers of occurrences in such reducts.

Table 2. Ordering of features based on analysis of reducts

Order 1 (QI_1)				Order 2 (QI_2)				Order 3 (QI_3)				Order 4			
of	3478			of	404.29			of	30915			and	6	144	
.	3190			.	379.17	M1		on	28040	M1		not	4	53	M1
on	3083	M1		on	351.08	M2		.	27929			by	3	108	M2
,	2943	M2		not	345.73			,	27517			from	3	90	
not	2778			and	340.67		L8	;	25137	M2		.	3	74	
;	2740	M3		by	330.02	M3		in	24359		L7	in	2	50	M3 L7
in	2726			,	323.63			this	23604	M3		if	1	29	M4
by	2648	M4	L9	in	317.50		L7	at	23293			:	1	23	
this	2585			;	307.53	M4	L6	not	23263		L6	on	1	9	L6
at	2585	M5	L8	at	297.20	M5		to	22443	M4		!	0	39	M5 L5
to	2497	M6		this	293.38			by	22114			?	0	26	M6 L4
:	2384		L7	to	287.72			:	21536		L5	to	0	18	M7
!	2368	M7		!	284.44		L5	!	20453	M5		of	0	16	
and	2324		L6	from	276.84	M6		with	19789			this	0	15	L3
from	2273	M8		:	273.81		L4	from	19613			-	0	14	M8
with	2161		L5	with	245.09	M7		as	19111			that	0	14	
as	2108			as	242.21			-	18415		L4	as	0	14	L2
-	2035		L4	-	233.49		L3	and	16204	M6		at	0	12	M9
?	1712	M9		?	195.36	M8		?	15688		L3	what	0	11	
for	1609		L3	for	184.74			for	14527	M7		(	0	10	L1
if	1584			if	183.39		L2	if	14378		L2	but	0	9	
what	1415		L2	what	163.50			(	13318			for	0	8	
(	1395			(	151.68			what	12787			with	0	6	
that	1343		L1	that	150.79		L1	that	12430		L1	,	0	5	
but	893			but	100.74			but	8212			;	0	3	

The orderings were used for reduction of the decision algorithm in the following way: when a feature was discarded, from the decision algorithm there were removed all rules which in their premise contained conditions on this feature.

Following the three new orderings of attributes, there were conducted 3 groups of tests. Any ordering can be read as increasing or decreasing, and it cannot be said for certain which is better, therefore, to ensure additional verification within each group of tests there were distinguished 2 subgroups: with keeping the attributes more important from a current perspective (denoted by "L"), and with discarding these more important features (denoted by "M"). The results present only the correct classification (treating ambiguous decisions as incorrect).

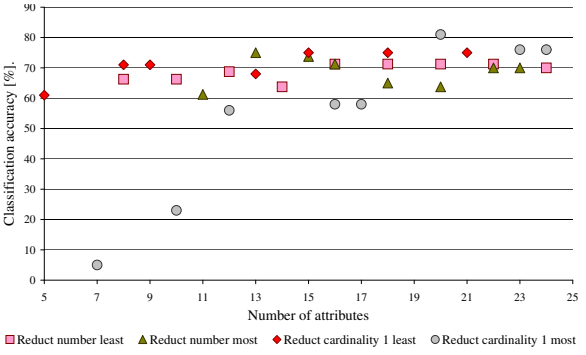


Fig. 1. DRSA classification accuracy in relation to the number of attributes with reduction based on Order 2 (shorter reducts are more important than long ones)

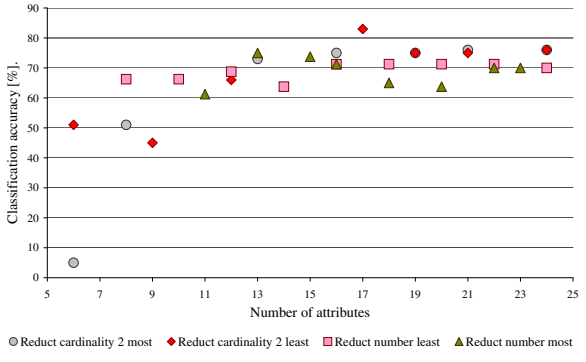


Fig. 2. DRSA classification accuracy in relation to the number of attributes with reduction based on QI_3 (shorter reducts are less important than long ones)

Through tests for each algorithm there was obtained such support, for which the accuracy was the highest, and these values are plotted in the graphs. For comparison there are given results for the past tests regarding only the number of reducts regardless of their cardinality.

The experiments show that when there is assumed greater significance of reducts with lower cardinalities (Fig. 1), disregarding the most significant features brings first some increase, then the significant decrease in accuracy. Keeping such features causes a slight and slow decline - still above 70% even when there are just 8 attributes left.

When longer reducts were assigned more influence (Fig. 2), removing the features most often present in such reducts preserved the classification ratio till there were 13 attributes left, after which the performance worsened. Keeping features less often present in longer reducts in the initial phase was similar, but when the number of attributes fell below 17, the classification ratio fell as well.

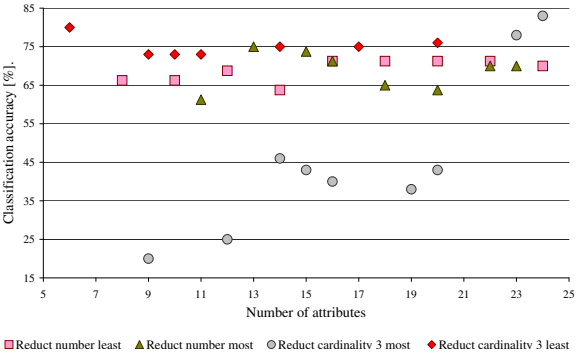


Fig. 3. DRSA classification accuracy in relation to the number of attributes with feature reduction based on Order 4 (focus on only the shortest reducts)

When there were considered only the shortest reducts (with cardinality 4 and 5) in Order 4 (Fig. 3), removing less important features while keeping these more important ones keeps the classification on almost the same level for the whole range of the decreasing number of attributes. The opposite approach brings quick and significant worsening of the performance.

For all orderings of attributes there exists at least one version of the decision algorithm that gives better results than the full algorithm. For Order 2: 529 rules involving 6 attributes were reduced to 36 with support≥40, with classification of 80%. For Order 3: 13559 rules with 17 attributes were reduced to 37 with support≥30, resulting in the classification accuracy of 83%. And for Order 4 the best case algorithm of 10,589 rules with 20 attributes, reduced to 96 with support≥20, gave 81% of correct classification.

For verification the reduced decision algorithms were tested (Table 3) for the testing set Test48, which gave 72% correct decisions for the full algorithm.

Table 3. Performance for the second testing set. Columns labelled N specify number of attributes, S states support for which the classification accuracy [%] was obtained for the two testing sets T1 (Test80) and T2 (Test48).

Order 2 M				Order 2 L				Order 3 M				Order 3 L				Order 4 M				Order 4 L			
N	S	T1	T2	N	S	T1	T2	N	S	T1	T2	N	S	T1	T2	N	S	T1	T2	N	S	T1	T2
24	40	76	72	24	40	76	72	24	40	76	72	24	40	76	72	24	30	83	72	20	40	76	72
23	40	76	72	21	40	75	75	21	40	76	72	21	40	75	75	23	28	78	64	17	40	75	75
20	20	81	62	18	40	75	75	19	34	75	72	19	40	75	75	20	10	43	41	14	40	75	75
17	12	58	60	15	40	75	75	16	34	75	75	17	30	83	72	19	4	38	25	11	40	73	75
16	12	58	60	13	26	68	68	13	17	73	77	12	22	66	64	16	4	40	29	10	40	73	75
12	7	56	60	9	22	71	68	8	2	51	64	9	5	45	50	15	4	43	33	9	40	73	75
10		23	25	8	22	71	70	6		5	8	6	3	51	22	14	3	46	39	6	40	80	75
7		5	8	5	15	61	60									12		25	29				
																9	2	20	22				

As two testing sets have their own characteristics, the results obtained cannot be identical, yet the general trends in the classification accuracy (observed for the same values of support required) indicate high degree of similarity, especially within L series of tests. Within M series there were several cases of differences between support values needed for the highest classification for these two sets.

6 Conclusions

When domain knowledge is insufficient, establishing the importance of individual features for the recognition can be attempted by exploiting the inherent properties of data mining techniques. The paper presents analysis of features for DRSA classifier applied in a stylometric task of authorship attribution. The attributes were studied from the perspective of the number of occurrences in constructed relative reducts, taking into account their cardinalities, which led to several orderings of attributes and shortened versions of decision algorithms. Experiments show that in all cases it was possible to obtain the significant reduction of data while keeping at least the same or even increasing classification accuracy.

Acknowledgments. 4eMka Software used in search for reducts and decision rules [3,7] was downloaded in 2008 from the website of Laboratory of Intelligent Decision Support Systems, (`http://www-idss.cs.put.poznan.pl/`), Poland.

References

1. Burrows, J.: Textual analysis. In: Schreibman, S., Siemens, R., Unsworth, J. (eds.) A companion to digital humanities. Blackwell, Oxford (2004)
2. Craig, H.: Stylistic analysis and authorship studies. In: Schreibman, S., Siemens, R., Unsworth, J. (eds.) A companion to digital humanities. Blackwell, Oxford (2004)
3. Greco, S., Matarazzo, B., Slowinski, R.: Dominance-based rough set approach as a proper way of handling graduality in rough set theory. Transactions on Rough Sets 7, 36–52 (2007)
4. Pawlak, Z.: Rough sets and intelligent data analysis. Information Sciences 147, 1–12 (2002)
5. Peng, R., Hengartner, H.: Quantitative analysis of literary styles. The American Statistician 56(3), 15–38 (2002)
6. Shen, Q.: Rough feature selection for intelligent classifiers. Transactions on Rough Sets 7, 244–255 (2006)
7. Słowiński, R., Greco, S., Matarazzo, B.: Dominance-based rough set approach to reasoning about ordinal data. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 5–11. Springer, Heidelberg (2007)
8. Stanczyk, U.: Dominance-based rough set approach employed in search of authorial invariants. In: Kurzynski, M., Wozniak, M. (eds.) Computer Recognition Systems 3. Advances in Intelligent and Soft Computing, vol. 57, pp. 293–301. Springer, Heidelberg (2009)
9. Stańczyk, U.: DRSA decision algorithm analysis in stylometric processing of literary texts. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 600–609. Springer, Heidelberg (2010)

Evolutionary Protein Contact Maps Prediction Based on Amino Acid Properties

Alfonso E. Márquez Chamorro, Federico Divina, and Jesús S. Aguilar-Ruiz

School of Engineering, Pablo de Olavide University of Sevilla, Spain
{amarcha,fdivina,aguilar}@upo.es

Abstract. Protein structure prediction is one of the main challenges in Bioinformatics. An useful representation for protein 3D structure is the protein contact map. In this work, we propose an evolutionary approach for contact map prediction based on amino acids physicochemical properties. The evolutionary algorithm produces a set of rules that identifies contacts between amino acids. The rules obtained by the algorithm imposes a set of conditions on four amino acid properties in order to predict contacts. Results obtained confirm the validity of the proposal.

Keywords: Protein Structure Prediction, Contact Map, Evolutionary Computation, Residue-residue Contact.

1 Introduction

Proteins are compounds composed of carbon, hydrogen, oxygen, and nitrogen, which are arranged as strands of amino acids. They are essential components of every live form, as they play an essential function, e.g., transport function, enzymatic function, structural function, etc. The specific biochemical function of a protein is consequence of its structure complexity, which is determined by the specific sequence of amino acids, as demonstrated by Anfinsen [1]. Being able to predict the three-dimensional structure of a protein from its sequence of amino acids, is one of the main open problems in Bioinformatics. Such a problem is known as Protein Structure Prediction (PSP). Solving this problem would allow to acquire the possibility of knowing the function of a protein directly from its amino acids sequence, and would represent a huge advance in various field, e.g., medical or biological areas.

The structure of a protein can be experimentally determined using techniques such as X-ray crystallography or resonance scans (NMR techniques). However, these techniques are expensive and time consuming.

In addition to this, more and more amino acids sequences of proteins are available by the day, but their three-dimensional structures remain often unknown, and so their functions cannot be determined. Consequently the gap between protein sequence information and protein structural information is increasing rapidly. It follows that computational methods are needed in order to reduce this gap, as they would provide a cheaper and faster way to solve the PSP problem.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 303–310, 2011.

We can identify three computational approaches for the prediction of 3D structures of proteins: homology-based models, threading models and *ab initio* models. As its name suggests, homology-based models predict protein structures based on sequence homology with known structures, e.g., [2,3]. The principle behind this is that if two proteins share a high degree of similarity in their sequences, then they should have similar 3D structures. Threading, or sequence-structure alignment methods try to determine the structure of a new protein sequence by finding its best "fit" to some fold in a library of structures, see, for instance [4]. *Ab initio* methods attempt to generate models of proteins solely based on sequence information and without the aid of known protein structures, e.g., [5,6]. Our proposal lies in this last category.

When a computational method is used, the first thing one has to do is selecting a way to encode the data. A common way for representing the three dimensional structure of a protein is provided by contact maps. A contact map (CM) is a two-dimensional representation of the tertiary structure of a protein. Contact maps are an useful and interesting approach for the representation of the structure of proteins since they capture all important feature of the folding process. A protein with an amino acid sequence of length N, can be represented by using a NxN symmetric matrix C. Each entry C_{ij} of C is either equal to 1 or 0, depending on whether or not there is a contact between amino acids i and j of the protein. In order to determine if two amino acids are in contact, a threshold μ, expressed in angstroms, is used. If the distance between amino acids i and j is less than μ, then we say that i and j are in contact.

In this paper, we propose a residue-residue contact map predictor based on an evolutionary algorithm (EA). The EA uses a set of representative amino acid properties in order to predict contacts between amino acids. We believe that EAs are suitable for solving the PSP problem, since PSP can be seen as a search problem through the space determined by all the possible foldings. Such a space is highly complex and has a huge size. Finding the optimal solution in such space is very hard. In these cases, EAs have proven to be effective methods that can provide sub-optimal solutions. The EA we propose, will produce a set of rules that express conditions on the particular biochemical properties of the amino acids. Such a model can then be used in order to determine whether or not there is a contact between amino acids. An advantage of such an approach is that the generated rules can easily be interpreted by experts in the field in order to extract further insight of the folding process of proteins.

Previously, EAs have been applied to PSP, e.g., [6], where Torsion angles representation has been used. Hydrophobic-Polar (HP) model and lattice model were employed in [5]. A contact map model generator was included in [4], while [7] used a bit encoding for proteins. Our evolutionary proposal is based on the incorporation of new amino acid properties which have not yet be considered for the prediction of protein structure.

The rest of the paper is organized as follows. In section 2, we describe our proposal to predict protein contact maps. Section 3 presents the experimentation performed in order to assess the validity of our proposal and a discussion of the

obtained results. Finally, in section 4, we draw some conclusions and analyze possible future work.

2 Methodology

In order to test our proposal, we have obtained a data set of protein structures from the Protein Data Bank (PDB) [8]. From this data set, we have then extracted physicochemical information about each protein. This data will be divided into a training set and a test set. The EA will use the training set in order to produce a set of contact prediction rules. From these rules, a contact map is built for each test protein sequence, which will be used to establish the accuracy of the obtained prediction model. This experimental procedure is illustrated in Figure 1.

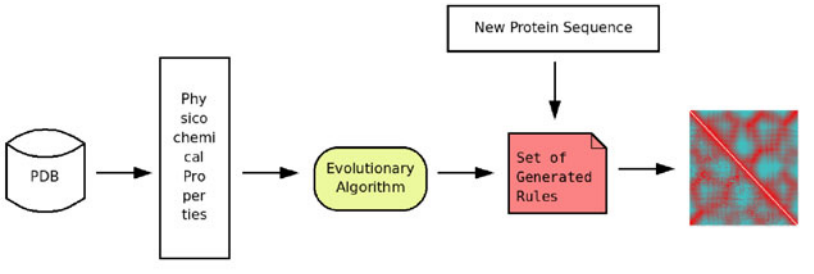


Fig. 1. Experimental and prediction procedure

As already mentioned, our EA will exploit a set of amino acids properties. A set of conditions on the selected properties will establish if a contact between two amino acids is predicted. It is known that amino acid properties have an important role in the PSP problem [9]. Several PSP methods were proposed that relied on amino acids properties, e.g., hydrophobicity and polarity were employed in HP models [5]. The prediction performed by our proposal will be based on four amino acids properties, in particular, hydrophobicity, polarity, charge and residue size.

In the following we address the various solutions adopted for what regards the representation, the genetic operators and the fitness function used by the EA.

2.1 Encoding

In our approach, an individual will encode a rule that determines if there is a contact between amino acid i and j. Such a rule will impose a set of conditions on the four properties of the amino acids. In order to do so, two windows of three amino acids are maintained, where one window is relative to amino acids $i-1, i, i+1$ and the other is relative to amino acids $j-1, j, j+1$.

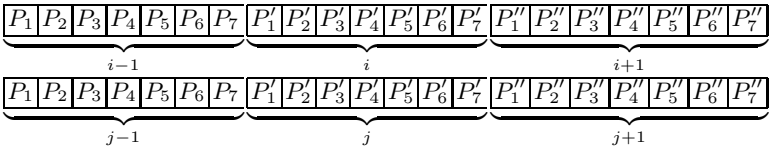


Fig. 2. Example of chromosome encoding for the $i-1$,i,$i+1$,$j-1$,j,$j+1$ residues

The encoding of the individuals is illustrated in figure 2. In the figure, positions P_1, P_2 represent the range of the hydrophobicity values for amino acid $i-1$. In the same way, positions P_3, P_4 represent the range of the polarity values, position P_5 encodes the net charge property value of the amino acid. Finally, positions P_6 and P_7 represent the range for the residue size of amino acid $i-1$. All these positions are encoded as real values.

We selected Kyte-Doolittle hydropathy profile for the hydrophobicity [10], the Grantham's profile [11] for polarity and Klein's scale for net charge [12]. The Dawson's scale [13] is employed to determine the size of the residues. The values of these properties are then normalized to a range between -1 and 1 for hydrophobicity, polarity and between 0 and 1 for the residue size. Three values are used to represent the net charge of a residue: -1 (negative charge), 0 (neutral charge) and 1 (positive charge).

2.2 Genetic Operators and Fitness Function

The algorithm starts with a randomly initialized population and the algorithm is run for a maximum of 100 generations. If the fitness of the best individual does not increase over twenty generations, the algorithm is stopped and a solution is provided. The population size is set to 100. In order to obtain the next generation, individuals are selected with a tournament selection mechanism of size two. Crossover and mutation are then applied in order to generate offspring. Elitism is also applied, therefore the fittest individual is always preserved in the next generation.

Various crossover operators have been tested. In particular, we have tested the performances of one-point, two-points, uniform and BLX-α crossovers. These crossover operators act at the level of the amino acid properties. So, for instance, the one-point crossover randomly select a point inside two parents and then builds the offspring using one part of each parent. For example, in figure 2, a possible point selected could be P_3 or P'_6. It follows that the resulting rule has to be tested for validity, since the so produced rule could contained incorrect ranges.

BLX-α crossover creates a new offspring $C = C_1, C_2 ... C_4 2$, where C_i is randomly selected in the interval $[C_{min} - I\alpha, C_{max} + I\alpha]$. C_{min} and C_{max} are the lower and higher values of the two parents at position i, and I is equal to $max_i - min_i$. An α value is also selected. In our case, we set the α value for the crossover in 0.1. This parameter must be higher or equal than 0. This crossover operator can be seen as a linear combination of the two parents. After having

performed several runs of the algorithm, the best results were obtained when the two-points crossover was used, which was then used as standard crossover in the algorithm.

Crossover operator and mutation are applied with a probability of 0.5. If mutation is applied, one gene of the individual is randomly selected, and its value is changed. If the selected gene is relative to the hydrophobicity, polarity, or residue size, its value is increased or decreased by 0.1. If the selected gene is relative to the net charge, its value is randomly changed to another allowed value $(-1, 0, 1)$. If the values of a mutated individual are not within the allowed ranges for each properties, the mutation is discarded.

The aim of the algorithm is to find both general and precise rules for identifying residue-residue contacts. Therefore, we have chosen as fitness function the F-measure, which is given by equation 1.

$$F = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision}. \tag{1}$$

Recall represents the proportion of identified contacts, while precision represents the error rate. The algorithm aims at maximizing the fitness.

At the end of the EA, the best rules are selected and they represent the solution. This is done in an incremental way: first, the best individual, according to its F-measure, is selected and added to the solution S. Then the next best individual is added to S, and the F-measure of S is calculated. This process is repeated until the addition of a rule causes the F-measure of S to decrease.

3 Experiments and Results

The experimentation has been performed with a data set described in [14]. This protein data set (56PDB) consists of 56 proteins with a sequence identity value lower than 25% and a sequence length lower than 100. As validation method we have used a 10-fold cross-validation.

Three statistical measures were computed to evaluate the accuracy of our algorithm:

- Recall represents the percentage of correctly identified positive cases. In our case, recall indicates what percentage of contacts have been correctly identified.
- Precision is a measure to evaluate the false positive rate. Precision reflects the number of correctly predicted examples.
- Specificity, or True Negative Rate, measures the percentage of correctly identified negative cases. In this case, specificity reflects what percentage of non-contacts have been correctly identified.

As already mentioned, an execution of the algorithm provides as a result a set of rules. If the algorithm is run several times, the final prediction model will consist of all the rules obtained at each execution. In other words, each time the algorithm is run, a number of rules are added to the final solution. Repeated

or redundant rules are not included in the final solution. As optimal and exact number of rules is unknown, we have performed various experiments varying the numbers of runs of the EA, where to a higher number of runs corresponds on higher number of rules. The aim of this was to test whether or not a higher number of rules would yield better results.

Table 1 shows the results for 100, 500, 1,000 and 2,000 executions. For instance, for 1,000 runs we have obtained 2,190 rules and for 2,000 runs we have finally obtained 4,866 rules.

Table 1. Average results and standard deviation obtained for different number of executions of the algorithm

Runs	$Recall_{\mu\pm\sigma}$	$Spec._{\mu\pm\sigma}$	$Prec._{\mu\pm\sigma}$	#rules
100	$0.144_{\pm 0.025}$	$0.993_{\pm 0.010}$	$0.502_{\pm 0.115}$	227
500	$0.539_{\pm 0.080}$	$0.985_{\pm 0.020}$	$0.462_{\pm 0.118}$	1,153
1000	$0.658_{\pm 0.130}$	$0.973_{\pm 0.025}$	$0.453_{\pm 0.125}$	2,190
2000	$0.683_{\pm 0.185}$	$0.965_{\pm 0.028}$	$0.438_{\pm 0.112}$	4,866

By examining table 1, we can notice that a low recall rate is achieved for 100 runs. However this result improves substantially when the number of rules in the final solution increases, reaching a maximum of 68% of correctly identified contacts for 2,000 runs. This is a consequence of having more rules in the final solution. In fact, the more rules in the final solution, the more cases are covered. The opposite holds for precision. In fact, it can be noticed that the precision obtained decreases with the increase of the number of rules in the final solution. This result was quite expected, since by covering more cases, the possibility of prediction errors increases due to the higher number of rules. Even in this case, the obtained precision is satisfactory in all the cases, as it never goes below 40%. We can also notice, that specificity is very satisfactory in all the settings, reaching values higher than 95% in all the cases. Notice that as for the precision, the specificity decreases when the number of rules increases. As far as the optimal number of rules is concerned, we can conclude that it is difficult to establish the optimal number of rules; with more rules, the true positive rate increases, however the false positive rate is also increased.

It is not possible to make an exact comparison between our method and the other existing methods. Each method uses a different structural data bases, different sets of proteins and different measures to evaluate the accuracy of their algorithms. However, other methods for PSP, e.g., [15], set the precision rate for a contact map prediction at about 30%. The precision obtained by our method clearly performed this result.

Figure 3 shows an example of a resulting rule. It can be seen that the rule has been divided into two windows of size three, as explained in section 2. If we inspect this rule, we can infer that, for example, the hydrophobicity value for the amino acid i lies between 0.52 and 0.92, the polarity value between -1.0 and -0.93, neutral charge (0.0), and a residue size between 0.77 and 0.97. Therefore,

the amino acid i could be L (Lysine) or F (Phenylalanine), which fulfills all these features according to the cited scales. As it can be noticed the produced rules are easily interpretable by experts in the field.

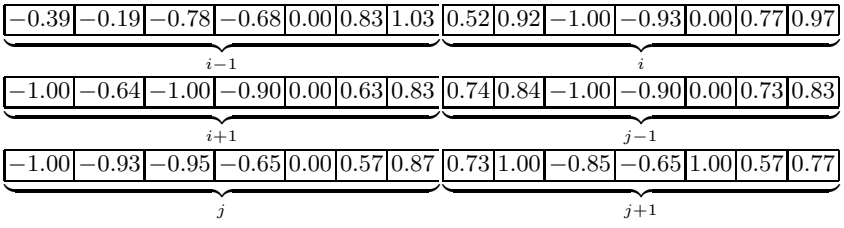


Fig. 3. Example of a resulting prediction rule

4 Conclusions and Future Work

In this article, we proposed an evolutionary algorithm approach for protein contact map prediction. Our algorithm generates a set of rules for residue-residue contact prediction using a representation based on four amino acid properties. Our approach can be used in an incremental way, in fact, the algorithm can be executed several times, and at each run, a set of rules is added to the final solution. This allows us to tune the results according to what measure we want to optimize. For example, by increasing the number of rules the recall increases. The rules forming the final solution express a set of conditions on the four phsysicochemical properties of amino acids. As a consequence, such rules can easily be interpreted and analyzed by experts in the field in order to obtain more insight on the protein folding process.

As for future work, we intend to expand this study to other significant amino acid properties, e.g., isoelectric point and steric parameter. Moreover, the length of window size of each individual could be variable, where the estimation of an adequate length would be determined by the evolutionary process. Another future development is the application of the algorithm to larger proteins data set, in order to test the validity of our proposal in these cases.

References

1. Anfinsen, C.: The formation and stabilization of protein structure. Biochem. J. 128, 737–749 (1972)
2. Zhang, Y.: I-tasser: fully automated protein structure prediction in casp8. Proteins: Structure, Function, and Bioinformatics 77, 100–113 (2009)
3. Tegge, A., Wang, Z., Eickholt, J., Cheng, J.: Nncon: Improved protein contact map prediction using 2d-recursive neural networks. Nucleic Acids Research 37(2), 515–518 (2009)
4. Gupta, N., Mangal, N., Biswas, S.: Evolution and similarity evaluation of protein structures in contact map space. Proteins: Structure, Function, and Bioinformatics 59, 196–204 (2005)

5. Unger, R., Moult, J.: Genetic algorithms for protein folding simulations. Biochim. Biophys. 231, 75–81 (1993)
6. Cui, Y., Chen, R.S., Hung, W.: Protein folding simulation with genetic algorithm and supersecondary structure constraints. Proteins: Structure, Function and Genetics 31, 247–257 (1998)
7. Zhang, G., Han, K.: Hepatitis c virus contact map prediction based on binary strategy. Comp. Biol. and Chem. 31, 233–238 (2007)
8. Protein data bank web, http://www.pdb.org
9. Gu, J., Bourne, P.E.: Structural Bioinformatics (Methods of Biochemical Analysis). Wiley-Blackwell, Chichester (2003)
10. Kyte, J., Doolittle, R.F.: A simple method for displaying the hydropathic character of a protein. J. J. Mol. Bio. 157, 105–132 (1982)
11. Grantham, R.: Amino acid difference formula to help explain protein evolution. J. J. Mol. Bio. 185, 862–864 (1974)
12. Klein, P., Kanehisa, M., DeLisi, C.: Prediction of protein function from sequence properties: Discriminant analysis of a data base. Biochim. Biophys. 787, 221–226 (1984)
13. Dawson, D.M.: The Biochemical Genetics of Man. In: Brock, D.J.H., Mayo, O. (eds.) (1972)
14. Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Prediction of contact map with neural networks and correlated mutations. Protein Engineering 14, 133–154 (2001)
15. Zhang, G.Z., Huang, D.S., Quan, Z.H.: Combining a binary input encoding scheme with rbfnn for globulin protein inter-residue contact map prediction. Pattern Recognition Letters 16(10), 1543–1553 (2005)

A Comparative Study between Two Regression Methods on LiDAR Data: A Case Study

Jorge García-Gutiérrez[1], Eduardo González-Ferreiro[2], Daniel Mateos-García[1], Jose C. Riquelme-Santos[1], and David Miranda[2]

[1] Department of Computer Languages and Systems, University of Seville, Reina Mercedes s/n, 41012 Seville, Spain
{jgarcia,mateos,riquelme}@lsi.us.es
[2] Land Laboratory - Department of Agroforestry Engineering, University of Santiago de Compostela, Campus Universitario s/n, 27002 Lugo, Spain
{eduardo.gonzalez,david.miranda}@usc.es

Abstract. Airborne LiDAR (Light Detection and Ranging) has become an excellent tool for accurately assessing vegetation characteristics in forest environments. Previous studies showed empirical relationships between LiDAR and field-measured biophysical variables. Multiple linear regression (MLR) with stepwise feature selection is the most common method for building estimation models. Although this technique has provided very interesting results, many other data mining techniques may be applied. The overall goal of this study is to compare different methodologies for assessing biomass fractions at stand level using airborne LiDAR data in forest settings. In order to choose the best methodology, a comparison between two different feature selection techniques (stepwise selection vs. genetic-based selection) is presented. In addition, classical MLR is also compared with regression trees (M5P). The results when each methodology is applied to estimate stand biomass fractions from an area of northern Spain show that genetically-selected M5P obtains the best results.

Keywords: Tasmanian blue gum, *Eucalyptus globulus*, remote sensing, regression trees, multiple linear regressions, stand biomass estimation.

1 Introduction

In order to guarantee forest sustainability, it is vital to consider both the economic and ecological functions of forests. Therefore, it is necessary to quantify existing resources for the strategic, tactical and operational planning of silvicultural treatments and forest operations. In the case of the forest biomass, it provides an indication of carbon sequestration in trees and an estimate of cellulosic material as a potential source of renewable energy [1].

Many forest management planning systems are based on the use of stand mean values of biophysical variables [2] often measuring in field. However, stand

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 311–318, 2011.

variables characterization and quantification methods are very expensive, time-consuming [3] and limited by the cost of establishing sufficient sample plots to capture the existing variability [4]. Furthermore, biomass estimation often involves destructive sampling [1]. In this context, the use of Airborne Laser Scanning (ALS), also referred to as Light Detection and Ranging (LiDAR), has been explored to reduce costs transforming the way change detection and forest mensuration is performed. LiDAR is a remote laser-based technology that can determine the distance from the source placed on an aerial platform to an object or surface providing not only X-Y position, but also the returned energy (laser intensity) and the coordinate Z for every impact. The distance to the object is determined by measuring the time between the pulse emission and detection of the reflected signal taking into account the position of the emitter.

A very important subset of forest applications like forest inventories [5], biomass estimation [6] or fuel models [7] are based on the estimation of variables in order to build models. If LiDAR is being used, those variables will usually be estimated by multiple linear regression (MLR) between field measurements and LiDAR metrics. The main advantage of using MLR is the simplicity of the resulting model. In contrast, the selected method also has some drawbacks: in most studies, the regression employs a suite of frequency-based metrics calculated from the previous LiDAR height and intensity data, which are systematically eliminated from a full model using a stepwise process which results in a set of predictors with little physical justification [8]. Thus, the methodologies to build regressions between some key variables for forest characterization and LiDAR data are being reviewed [9]. Moreover, new non-parametric techniques and genetic algorithms applied to the predictor selection [10] have been used [11] improving the results but also losing part of the linear regression model's simplicity and clarity.

To the best of our knowledge, the joint use of genetic algorithms and regression trees has not been accurately exploited in the context of biomass estimation since they can maintain the simplicity of stepwise-selected MLR improving its performance. Thus, two comparisons are presented in this work. First, the traditional stepwise selection is compared with a genetic feature selection. Then, a comparison between MLR and a M5P regression tree [12] both genetically-selected is also proposed. These comparisons aim to fulfill three objectives:

- Show the higher level of accuracy when genetic algorithms are applied in lieu of the classical stepwise feature selection.
- Show the improvement on the regression quality when more complex data mining techniques such as M5P replace MLR.
- Establish a solid study to back the exploration of new improvements in regression trees in order to enhance LiDAR products.

The rest of the paper is organized as follows. Section 2 provides a description of the real data used in this work, highlighting the final selected features. Section 3 describes the methodology used. The results achieved are shown in Section 4 and, finally, Section 5 is devoted to summarising the conclusions and to discussing future lines of work.

2 Data Description

The study area is located in the north of Galicia in the northwest of Spain (see Figure 1). The LiDAR data covered 4 km^2 of high density *Eucalyptus globulus* plantations and were acquired in November 2004.

Fig. 1. Image of the study area located in Trabada in the northern region of Galicia (Spain). In blue, the areas flown. In green, the centroids of the inventory plots.

A forest inventory of 39 square plots of 15 m^2 was conducted in mature *Eucalyptus globulus* plantations in February and March 2005. From that fieldwork, crown biomass (W_{cr}), stem biomass (W_{st}) and aboveground biomass (W_{abg}) were calculated.

A set of common metrics in literature [13,14,15] were calculated from the normalized intensity and height values of LiDAR data collected within the limits of the 39 field plots. These metrics are used as independent variables in the regression models whilst W_{cr}, W_{st} and W_{abg} are selected as dependent variables.

3 Method

In order to select the best predictors, a genetic feature selection from LiDAR metrics is carried out. A deeper description of the genetic algorithm (GA) and its characteristics is provided in the following paragraphs along with a brief description of the types of regression used in this study.

3.1 Initial Population

To execute the genetic algorithm, an individual representation is required. In this case, an individual of the population is an array whose cells each represent a weight for each feature in the training set. Each weight is initialized with a value of 1 or 0. Thus, if the corresponding feature is selected, the weight will be 1, otherwise 0.

The size of the population and the number of generations are genetic algorithm parameters which are set up with values of 200 and 100 respectively in this work.

These parameters were empirically selected and proved to reach the best results. In addition, every simple linear regression is part of the initial population which involves to start from the best possible minimum model.

3.2 Fitness Function

The fitness function for discriminating individuals who best fit each generation is based on the coefficient of determination R^2 which measures the adjustment with the training data. This value fluctuates between 0 and 1. The higher R^2, the better the individual.

A related problem with the simplicity of regression models is multicollinearity. The control of this detrimental effect is performed using the condition number as a threshold. The condition number is associated with the eigenvalues of the matrix built by the features selected in the individual. Moreover, it is well-known that a condition number that exceeds a value of 30 involves a high degree of multicollinearity. In this way, every individual with a condition number of 30 or higher is assigned a fitness value of 0.

3.3 Crossover and Mutation

In the design of a GA, it is always important to establish a coherent search criterion in the space of possible solutions. This can only be achieved with a proper selection of crossover and mutation operators.

A random crossover operation for two individuals (parents) selected by the roulette-wheel method is applied. The crossover selects a gen (weight) for each feature from two possible values (the parents values associated to the corresponding feature) randomly. In the end, the final set of genes is assigned to the new individual.

The mutation operator has been defined to change the value of a weight according to a probability.In our case a value of 0.1 was empirically selected. A mutation involves changing a gen value for its complementary (1 into 0 and vice versa).

3.4 Regression Models

Linear and allometric models were used to establish empirical relationships between field measurements and LiDAR variables. Their general expressions can be seen in Equation 1 and 2 respectively.

$$Y = \beta_0 + \beta_1 \, X_1 + \beta_2 \, X_2 + \ldots + \beta_n \, X_n \tag{1}$$

$$Y = \beta_0 \, X_1^{\beta_1} \, X_2^{\beta_2} \, \ldots \, X_n^{\beta_n} \tag{2}$$

where Y are field values of W_{cr} (kg ha^{-1}), W_{st} (kg ha^{-1}), and W_{abg} (kg ha^{-1}), and $X_1, X_2, \ldots X_n$ may be variables related to the metrics of heights and pulse intensities distributions or measurements related to canopy closure [13].

4 Results

In this work, two well-defined comparisons are proposed. In the first, we compare classical stepwise and genetic-based feature selections. In the second, after genetically selecting the best predictors, a comparison between the classical MLR and M5P regression tree is established. For both methods, we use the WEKA framework [16] to generate the results.

As mentioned previously, checking whether the classical stepwise process on the possible predictors is enhanced by a genetically-based feature selection is one of the main objectives of this work. Due to the random nature of genetic algorithms and in order to establish the comparison, the execution of the genetic algorithms was repeated thirty times and the averaged values were taken. In Table 1, the selected predictors collected by each method can be seen beside the coefficient of determination R^2 achieved by the MLR technique for each case.

For every biomass dependent variable (W_{cr}, W_{st}, W_{abg}) an improvement is reached for both allometric and linear models when genetic selection is applied.

Table 1. Prediction capacity (coefficient of determination, R^2) for MLR when a stepwise and a genetic selection are respectively applied

Variable	Stepwise R^2	Stepwise predictors	Genetic R^2	Genetic predictors
allometric W_{cr}	0.619	h_{60}	0.759	h_{75}
allometric W_{st}	0.740	h_{60}	0.863	h_{75}
allometric W_{abg}	0.727	h_{60}	0.853	h_{75}
linear W_{cr}	0.708	h_{90}	0.753	h_{90},i_{mode},i_{ID}
linear W_{st}	0.801	h_{SKw},h_{75}	0.814	h_{SKw},h_{75}, i_{mode}, i_{70}
linear W_{abg}	0.771	h_{95}	0.809	h_{min},h_V,h_{75}

The next step consists in comparing the classical MLR and M5P regression tree generator when a GA is applied to make the feature selection. It is important to outline that the fitness function chosen for the GA optimizes the use of MLR so M5P regression tree starts with some disadvantage. Anyway, as seen in Table 2, M5P gets the same value of R^2 as MLR in the worst case, overcoming MLR in two out of six tests.

To statistically validate the differences between MLR and M5P, a test of statistical significance is needed. Since the real data is too small (just 39 instances in only one dataset), the study has to be built from other sources. Thus, the 10-fold cross-validation results on 27 well-known datasets [17] are collected. Once the coefficients of determination for every dataset are obtained for both methods (see Table 3), it is possible to establish a statistical analysis of their prediction capacity. Traditionally, parametric statistical tests such as ANOVA are applied for this type of analysis. However, for a comparison of these types of tests to be correct, the data must meet the criteria of independence, normality, and homoscedasticity [18]. Through a D'Agostino-Pearson test [19], it could thus be

Table 2. Prediction capacity (coefficient of determination, R^2) of genetically-selected MLR and M5P respectively

Variable	MLR averaged R^2	M5P averaged R^2
allometric W_{cr}	0.759	0.759
allometric W_{st}	0.863	0.863
allometric W_{abg}	0.853	0.853
linear W_{cr}	0.753	0.753
linear W_{st}	0.814	0.820
linear W_{abg}	0.809	0.826

confirmed that the data obtained for this study did not meet the criteria of normality. For this reason, a non-parametric approximation (Wilcoxon test) was selected [20]. The p-value results in a value less than 0.0001 so it can be said that differences between the methods are statistically significative (at $\alpha = 0.05$).

Having found that the number of wins is higher for M5P (5 for MLR and 12 for M5P) and knowing their differences are statistically significative, we can conclude that M5P outperforms MLR.

Table 3. Prediction capacity (coefficient of determination, R^2) of genetically-selected MLR and M5P respectively when both methods are applied to 27 datasets

Dataset	MLR R^2	M5P R^2	Dataset	MLR R^2	M5P R^2
auto93.arff	0.631	0.631	autoHorse.arff	0.801	0.801
autoMpg.arff	0.698	0.736	autoPrice.arff	0.808	0.823
bodyfat.arff	0.976	0.978	breastTumor.arff	0.000	0.000
cholesterol.arff	0.049	0.044	echoMonths.arff	0.124	0.124
housing.arff	0.636	0.830	hungarian.arff	0.302	0.298
kdd_coil_train1.arff	0.298	0.470	kdd_coil_train2.arff	0.164	0.164
kdd_coil_train3.arff	0.115	0.115	kdd_coil_train5.arff	0.114	0.114
kdd_coil_train6.arff	0.120	0.120	kdd_coil_train7.arff	0.066	0.066
kdd_el_ninosmall.arff	0.793	0.811	machine_cpu.arff	0.865	0.946
meta.arff	0.110	0.075	pbc.arff	0.266	0.305
pharynx.arff	0.000	0.000	pyrim.arff	0.752	0.718
quake.arff	0.006	0.040	stock.arff	0.532	0.746
strike.arff	0.098	0.234	triazines.arff	0.318	0.487
wisconsin.arff	0.219	0.209			

5 Conclusions

LiDAR technology has become an important tool for carrying out several important tasks for the natural environment and, in particular, for biomass estimation. Lately, biomass estimation models have been built by means of LiDAR data processing. In this work, two different comparisons were established when regression techniques were applied to LiDAR data. First, a comparison between a genetically-based and a classical stepwise feature selection was presented. The

study concluded that the GA outperformed the stepwise process when MLR was built using each set of selected predictors. Then, from a genetically-based feature set, two regression methods were tested: classical MLR and M5P regression trees. In this case, the results showed that M5P obtained better results when both methods were applied to real data from Galicia (Spain).

According to the results, new intelligent techniques applied to regression trees can be explored to improve the results when applied to biomass estimation. With this purpose, evolutionary computation could be used to overcome some M5P limits, optimizing the predictor selection and controlling the thresholds of the regression tree branches. Furthermore, a more in-depth comparison of regression trees with other non-parametric methodologies (support vector machines, neural networks) is required. Finally, an important aspect not explored in this work is the ability of regression trees for detecting the most important predictors (regression tree roots) which should be developed in future research.

References

1. Popescu, S.C.: Estimating biomass of individual pine trees using airborne lidar. Biomass and Bioenergy 31, 646–655 (2007)
2. Naesset, E., Gobakken, T., Holmgren, J., Hyyppä, H., Hyyppä, J.: Laser scanning of forest resources: the nordic experience. Scand. J. Forest. Res. 19, 482–499 (2004)
3. Hall, S., Burke, I., Box, D., Kaufmann, M., Stoker, J.: Estimating stand structure using discrete-return lidar: an example from low density, fire prone ponderosa pine forests. Forest. Ecol. Manag. 208, 189–209 (2005)
4. Lovell, J., Jupp, D., Newnham, G., Coops, N., Culvenor, D.: Simulation study for finding optimal lidar acquisition parameters for forest height retrieval. Forest. Ecol. Manag. 214, 398–412 (2005)
5. Anderson, J.E., Plourde, L.C., Martin, M.E., Braswell, B.H., Smith, M.L., Dubayah, R.O., Hofton, M.A., Blair, J.B.: Integrating waveform lidar with hyperspectral imagery for inventory of a northern temperate forest. Remote Sensing of Environment 112(4), 1856–1870 (2008)
6. Garcia, M., Riano, D., Chuvieco, E., Danson, F.M.: Estimating biomass carbon stocks for a Mediterranean forest in central Spain using LiDAR height and intensity data. Remote Sensing of Environment 114(4), 816–830 (2010)
7. Mutlu, M., Popescu, S.C., Stripling, C., Spencer, T.: Mapping surface fuel models using lidar and multispectral data fusion for fire behavior. Remote Sensing of Environment 112(1), 274–285 (2008)
8. Muss, J.D., Mladeno, D.J., Townsend, P.A.: A pseudo-waveform technique to assess forest structure using discrete lidar data. Remote Sensing of Environment (2010) (in Press)
9. Salas, C., Ene, L., Gregoire, T.G., Næsset, E., Gobakken, T.: Modelling tree diameter from airborne laser scanning derived variables: A comparison of spatial statistical models. Remote Sensing of Environment 114(6), 1277–1285 (2010)
10. Gong, B., Im, J., Mountrakis, G.: An artificial immune network approach to multisensor land use/land cover classification. Remote Sensing of Environment 115(2), 600–614 (2011)
11. Latifi, H., Nothdurft, A., Koch, B.: Non-parametric prediction and mapping of standing timber volume and biomass in a temperate forest: Application of multiple optical/LiDAR-derived predictors. Forestry 83(4), 395–407 (2010)

12. Quinlan, R.J.: Learning with continuous classes. In: 5th Australian Joint Conference on Artificial Intelligence, pp. 343–348 (1992)
13. González-Ferreiro, E., Diéguez-Aranda, U., Gonçalves-Seco, L., Crecente, R., Miranda, D.: Assessing biomass in Eucalyptus globulus plantations in Galicia using different LiDAR sampling densities. In: Miranda, D., Suárez, J., Crecente, R. (eds.) Proceedings of ForestSat 2010: 4th international conference on Operational tools in forestry using remote sensing techniques, Lugo, Spain, September 6-10, pp. 37–41 (2010)
14. Antonarakis, A., Richards, K., Brasington, J.: Object-based land cover classification using airborne LIDAR. Remote Sensing of Environment (112), 2988–2998 (2008)
15. Hudak, A.T., Crookston, N.L., Evans, J.S., Halls, D.E., Falkowski, M.J.: Nearest neighbor imputation of species-level, plot-scale forest structure attributes from LIDAR data. Remote Sensing of Environment 112, 2232–2245 (2008)
16. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1) (2009)
17. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
18. Zar, J.: Biostatistical Analysis. Prentice-Hall, Englewood Cliffs (1999)
19. Trujillo-Ortiz, A., Hernandez-Walls, R.: DagosPtest: D'Agostino-Pearson's K2 test for assessing normality of data using skewness and kurtosis. A MATLAB file (2003)
20. Cardillo, G.: Wilcoxon test: non parametric wilcoxon test for paired samples (2006)

Analysis of Measures of Quantitative Association Rules

M. Martínez-Ballesteros and J.C. Riquelme

Department of Computer Science, University of Seville, Spain
{mariamartinez,riquelme}@us.es

Abstract. This paper presents the analysis of relationships among different interestingness measures of quality of association rules as first step to select the best objectives in order to develop a multi-objective algorithm. For this purpose, the discovering of association rules is based on evolutionary techniques. Specifically, a genetic algorithm has been used in order to mine quantitative association rules and determine the intervals on the attributes without discretizing the data before. The algorithm has been applied in real-word climatological datasets based on Ozone and Earthquake data.

Keywords: Data mining, evolutionary algorithms, quantitative association rules.

1 Introduction

The use of massive processing techniques has revolutionized the scientific research and it has highly increased the amount of data obtained. Data mining is the most used instrumental tool in discovering knowledge from transactions.

In this context we present the result of applying a data mining technique, specifically, association rules (ARs), to data from several experiments. The aim of this process of mining ARs is discover the presence of pairs (attribute (A) - value (v)), which appear in a dataset with certain frequency in order to formulate the rules that display the existing relationship among the attributes.

A revision of the published literature reveals that there are many algorithms to find these rules. Most of the association rule (AR) algorithms are based on methods proposed by Agrawal et al. such as AIS [1] and Apriori [16], SETM[11], etc. Many tools that work in continuous domains just discretize the attributes using a specific strategy and treating these attributes as if they were discrete. Many others are based on evolutionary algorithms. Genetic Algorithms (GAs) [10] are used to solve AR problems because they offer a set of advantages for knowledge extraction and specifically for rule induction processes. Authors of [14] proposed a genetic algorithm (GA) to discover numeric ARs, dividing the process in two phases. Another GA was used in [17] in order to obtain quantitative ARs and confidence was optimized in the fitness function.

Some researches tried to visualize AR mining as a multi-objective problem rather than a single objective. Therefore, several measures can be considered as

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 319–326, 2011.

an objective. In [3] a multi-objective pareto-based GA was presented where the fitness function was formed by four different objectives.

In preliminary work [12][13], authors of this paper developed several single-objective GA that use a weighting scheme for the fitness function which involved some evaluation measures. It is known that a scheme of this nature is not ideal compared to multi-objective schemes, so that could reduce the features used in the fitness function for applying a multi-objective technique. So we expected to extend these algorithms to multi-objective algorithms. However the problem arises when choosing the right objectives to optimize the condition being treated.

Thus, the main motivation of this paper is to analyze the relationship among different evaluation measures of the ARs, in order to classify them in positively correlated, negatively correlated or not correlated. The study is the first step to select the best objectives involved in the subsequent development of a multiobjective GA for extracting ARs. To carry out the study a GA [12] is used for mining quantitative ARs. The algorithm has been applied in two real-world datasets, concretely in ozone data and earthquake data.

The rest of the paper is organized as follows. Section 2 provides a brief preliminary on ARs and some interestingness measures proposed in the literature. Section 3 describes an introduction of multi-objective algorithms. The results obtained are discussed in Section 4. Finally, Section 5 provides the achieved conclusions.

2 Association Rules

In the field of data mining and machine learning, ARs are used to discover common events in a dataset. Several methods have been extensively researched for learning ARs that have been proven to be very interesting to discover relationships among variables in large datasets [16][2]. ARs are classified as unsupervised learning in machine learning.

The AR mining finds interesting associations and/or correlation relationships among elements of large datasets. A typical example is the market-basket analysis [1]. In addition they are widely used in other many fields. It is also useful in the healthcare environment to identify risk factors in the onset or complications of diseases. This form of knowledge extraction is based on statistical techniques such as correlation analysis and variance. One of the most widely used algorithms is the Apriori algorithm.

Formally, an AR is a relationship among attributes in a dataset in the way $A \Rightarrow B$, where A and B are pair conjunctions such as $A = v$ if $A \in \mathbb{Z}$ or $A \in [v_1, v_2]$ if $A \in \mathbb{R}$. Generally, the antecedent A is formed by the conjunction of multiple pairs and the consequent B is usually a single pair.

2.1 Interestingness Measures for Association Rules

The following paragraphs detail the popular measures used to characterize an AR. It is important evaluate the quality of the rule in order to select the best ones and evaluate the results obtained.

Support(A)[9]: The support of an itemset A is defined as the ratio of transactions in the dataset that contain A. Usually, the support of A is named as the probability of A.

$$sup(A) = P(A) = \frac{n(A)}{N}. \tag{1}$$

where $n(A)$ is the number of occurrences of antecedent A in the dataset, and N is the number of transactions forming such dataset.

Support($A \Longrightarrow B$)[9]: The support of the rule $A \Longrightarrow B$ is the percentage of transactions in the dataset that contain A and B simultaneously.

$$sup(A \Longrightarrow B) = P(A \cap B) = \frac{n(AB)}{N}. \tag{2}$$

where $n(AB)$ is the number of instances that satisfy the conditions for the antecedent A and consequent B simultaneously.

Confidence($A \Longrightarrow B$)[9]: The confidence is the probability that transactions containing A, also contain B. In other words, it is the support of the rule divided by the support of the antecedent.

$$conf(A \Longrightarrow B) = \frac{sup(A \Longrightarrow B)}{sup(A)} \tag{3}$$

Lift($A \Longrightarrow B$)[4]: Lift or interest is defined as how many times A and B are together in the dataset more often than expected, assuming that the presence of A and B in transactions are occurrences statically independent. Lift greater than one involves statistical dependence in simultaneous occurrence of A and B. In other words, the rule provides valuable information about A and B occurring together in the dataset.

$$lift(A \Longrightarrow B) = \frac{P(A \mid B)}{P(B)} = \frac{sup(A \Longrightarrow B)}{sup(A)sup(B)} = \frac{conf(A \Longrightarrow B)}{sup(B)} \tag{4}$$

Conviction($A \Longrightarrow B$)[4]: Conviction was introduced as an alternative to confidence for mining ARs in relational databases. Values in the range (0, 1) means negative dependence, higher than 1 means positive dependence and a value equals to 1 means independence. Conviction is directional and gets its maximum value (infinity) when the implication is perfect, that is, if whenever A occurs also happens B.

$$conv(A \Longrightarrow B) = \frac{P(A)P(\neg B)}{P(A \cap \neg B)} = \frac{sup(A)sup(\neg B)}{sup(A \Longrightarrow \neg B)} = \frac{1 - sup(B)}{1 - conf(A \Longrightarrow B)} \tag{5}$$

Gain($A \Longrightarrow B$)[9]: Gain is calculated from the difference between the confidence of the rule and consequent support. It is also known as added value or change of support.

$$Gain(A \Longrightarrow B) = P(A \mid B) - P(B) = conf(A \Longrightarrow B) - sup(B) \tag{6}$$

Certainty Factor($A \Longrightarrow B$)**[8]:** Certainty factor was introduced by Shortliffe and Buchanan to represent uncertainty in the MYCIN expert system. It is a measure of the variation of the probability that B is in a transaction when we consider only those transactions where A is. A similar interpretation can be done for negative CFs. The certainty factor takes values in [-1, 1] and achieves its maximum possible value, 1, if and only if the rule is totally accurate.
$Conf(A \Longrightarrow B) > Sup(B)$

$$CF(A \Longrightarrow B) = \frac{P(A \mid B) - P(B)}{1 - P(B)} = \frac{conf(A \Longrightarrow B) - sup(B)}{1 - sup(B)} \quad (7)$$

$Conf(A \Longrightarrow B) <= Sup(B)$

$$CF(A \Longrightarrow B) = \frac{P(A \mid B) - P(B)}{P(B)} = \frac{conf(A \Longrightarrow B) - sup(B)}{sup(B)} \quad (8)$$

Leverage($A \Longrightarrow B$) **[15]:** Leverage measures the proportion of additional cases covered by both A and B above those expected if A and B were independent of each other. Values above 0 are desirable. In addition, leverage is a lower bound for support, so optimizing only leverage guarantees a certain minimum support (contrary to optimizing only confidence or only lift).

$$lev(A \Longrightarrow B) = P(A \cap B) - P(A)P(B) = sup(A \Longrightarrow B) - sup(A)sup(B) \quad (9)$$

In most cases, it is sufficient to focus on a combination of support, confidence, and either lift or leverage to quantitatively measure the "quality" of the rule. However, the real value of a rule, in terms of usefulness and actionability is subjective and depends heavily of the particular domain and business objectives.

3 Multi-objective Optimization

GAs are search algorithms which generate solutions to optimization problems using techniques inspired by natural evolution [10]. They are implemented as a computer simulation in which a population of abstract representations (chromosomes) of candidate solutions (individuals) to an optimization problem evolves toward better solutions. In this context, a classical real-coded GA (RCGA) is used due to the domain of the ARs is continuous, thus, the algorithm deals with numeric data during the whole rule extraction process.

Evolutionary algorithms were originally designed for solving single objective optimization problems. However, many real world optimization problems have more than one objective in conflict with each other. Since multi-objective optimization searches for an optimal vector (rules in data mining) an not just a single value (one rule), one solution often cannot be said to be better than another and there exists not only a single optimal solution, but a set of optimal solutions, called the Pareto-optimal set [19]. The presence of multiple conflicting objectives and the need of using decision-making principles cause a number of different problem scenarios to emerge in practice.

In the last two decades an increasing interest has been developed in the use of GAs for multiobjective optimization. There are multiple proposals of multi-objective GAs [5] as the algorithms MOGA [7], NSGA II [6] or SPEA2 [18] for instance.

The mining process of ARs can be considered as a multi-objective problem rather than a single objective one, in which the measures used for evaluating a rule can be thought as different objectives. There are two goals in multi-objective optimization in the mining of ARs. First, discover rules as close to the Pareto-optimal as possible, and second, find rules as diverse as possible in the obtained non-dominated set. For this purpose, it is necessary to define the best objectives in order to get rules with high accuracy, comprehensible and interesting. In this proposal, several experiments have been carried out and the results are shown in Section 4. The aim of this study is to analyze the correlation and relationships among different evaluation measures of the ARs to define the objectives in order to design a multi-objective GA in this context.

4 Experimental Study

Several experiments have been carried out in this paper to evaluate the relationship among different interestingness measures of ARs. As a preliminary step, the proposed algorithm in [12] by the authors of this work was applied in order to achieve the AR mining task. Two kind of real-world datasets are considered in this work to prevent the resulting set of measures are not dependent on the datasets:

- Ozone concentration: Four datasets have been used containing a compact monthly average values including total ozone content (TOC), over different sites at Iberian Peninsula: Madrid, Arenosillo, Lisbon and Murcia. TOC series are based on ozone data from the Total Ozone Mapping Spectrometer (TOMS) on board the NASA Nimbus-7 satellite from 1st November 1978 to 6th May 1993. Each dataset consists of eight quantitative attributes and 172 samples.
- Earthquakes: The earthquake dataset was collected from the catalogue of Spanish's Geographical Institute (SGI). This dataset consists of four attributes related to location and magnitude of Spanish earthquakes from 1981 to 2008 and 873 samples.

Afterwards, the interestingness measures described in Subsection 2.1 were calculated for the quantitative ARs obtained by the algorithm for each dataset and included in a single database. A statistical study has been carried out to analyze the relationships and dependencies among measures. Specifically, correlation coefficient and principal component analysis (PCA) was applied among the measures.

Correlation coefficient is a measure of the correlation (linear dependence) between two variables X and Y, giving a value between +1 and -1 inclusive. Correlation is +1 in the case of a perfect positive (increasing) linear relationship

(correlation), -1 in the case of a perfect decreasing (negative) linear relationship (anticorrelation) [5], and some value between -1 and 1 in all other cases, indicating the degree of linear dependence among the variables. The closer the coefficient is to either -1 or 1, the stronger the correlation among the variables.

PCA is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components.

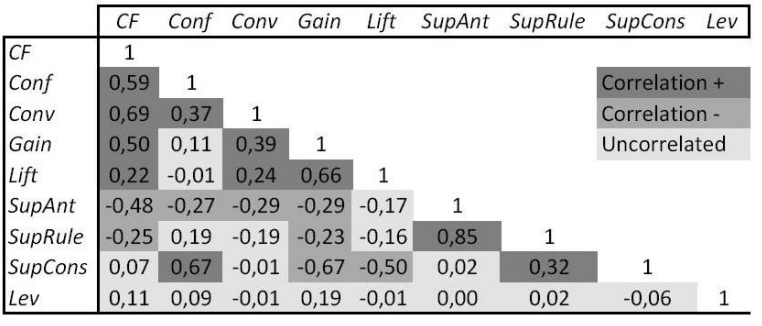


Fig. 1. Correlation coefficients

Table 1. Rotated Components

Measure	Component 1	Component 2	Component 3	Component 4
CF	**0,837**	0,228	-0,276	0,094
Conf	**0,887**	-0,283	0,081	0,101
Conv	**0,721**	0,308	-0,149	-0,120
Gain	0,308	**0,862**	-0,138	0,189
Lift	0,166	**0,808**	-0,019	-0,089
SupAnt	-0,322	-0,035	**0,913**	-0,008
SupRule	0,057	-0,164	**0,973**	0,024
SupCons	0,434	**-0,857**	0,164	-0,050
Lev	0,037	0,052	0,015	**0,985**

Figure 1 shows a table of correlation coefficient among measures: certainty factor (CF), confidence ($conf$), conviction ($conv$), gain, lift, support of antecedent ($supAnt$), support of rule ($supRule$), support of consequent ($supCons$) and leverage (lev). In the table three cases of correlation have been distinguished: Positive correlation (correlation +) when the coefficient is greater than 0.2, negative correlation (correlation -) when the coefficient is less than -0.2, and not correlation (uncorrelated) in other case. Some interesting conclusions can be extracted from these results.

It can be observed that CF is positively correlated to $conf$, $conv$, $gain$ and $lift$, and negatively correlated to $supAnt$ and $supRule$. CF and $conf$ are the measures that best correlates positively with other measures. Also, $supRule$ is

strong correlated with *supAnt*. *supCons* is correlated with *gain*, $lift$ and $conf$. However, *lev* is uncorrelated with other measures, thus, independent of other measures.

Table 1 presents the matrix of components with PCA as extraction method and Varimax with Kaiser Normalization as rotation method. The aim of this table is to find groups among the measures, and select the best representative for each group. This study may be useful to choose the various objectives that could be optimized in a multi-objective algorithm for mining ARs. It can be noticed that there are four principal components corresponding each to an independent group of measures. CF, $conf$ and *conv* belongs to the first group because they are the most correlated in the *Component* 1. *Gain*, $lift$ and *supCons* belongs to the second group due to the highest correlation in the *Component* 2. The third group contains *supAnt* and *supCons* and finally, *lev* is only measure of the group 4. In order to select the best objectives, we can study the most correlated for each group. Therefore, $conf$, *gain*, *supRule* and *lev* could be good candidates to optimize the mining of ARs by a multi-objective algorithm.

5 Conclusions

A method of analysis of quality measures of ARs has been proposed in this work. The ARs mining process can be considered as a multi-objective problem rather than a single objective. However, the selection of the best objectives candidates is not arbitrary. Several experiments have been carried out in order to analyze the relationship among different evaluation measures as a previous step before implementing a multi-objective algorithm for association rules. The results have determined that correlation coefficient and principal component analysis can be useful to define dependencies and grouping the interestingness measures of ARs.

Acknowledgments. The financial support from the Spanish Ministry of Science and Technology, project TIN2007-68084-C-00, and from the Junta de Andalucía, project P07-TIC-02611, is acknowledged.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)
2. Alatas, B., Akin, E.: An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. Soft Computing 10(3), 230–237 (2006)
3. Alatas, B., Akin, E., Karci, A.: MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules. Applied Soft Computing 8(1), 646–656 (2008)
4. Brin, S., Motwani, R., Silverstein, C.: Beyond market baskets: generalizing association rules to correlations. In: Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, vol. 26, pp. 265–276 (1997)

5. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc., Chichester (2001)
6. 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)
7. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416–423. Morgan Kaufmann, San Francisco (1993)
8. Fu, L.M., Shortliffe, E.H.: The application of certainty factors to neural computing for rule discovery. IEEE Transactions on Neural Networks 11(3), 647–657 (2000)
9. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Comput. Surv. 38(3), 9 (2006)
10. Goldberg, E.D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, Reading (1989)
11. Houtsma, M., Swami, A.: Set-Oriented Mining for Association Rules in Relational Databases, pp. 25–33. IEEE Computer Society Press, Los Alamitos (1995)
12. Martínez-Ballesteros, M., Martínez-Álvarez, F., Troncoso, A., Riquelme, J.C.: Quantitative association rules applied to climatological time series forecasting. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 284–291. Springer, Heidelberg (2009)
13. Martínez-Ballesteros, M., Martínez-Álvarez, F., Troncoso, A., Riquelme, J.C.: Mining quantitative association rules based on evoluationary computation and its application to atmospheric pollution. Integrated Computer-Aided Engineering 17(3), 227–242 (2010)
14. Mata, J., Alvarez, J.-L., Riquelme, J.-C.: Discovering numeric association rules via evolutionary algorithm. In: Chen, M.-S., Yu, P.S., Liu, B. (eds.) PAKDD 2002. LNCS (LNAI), vol. 2336, p. 40. Springer, Heidelberg (2002)
15. Piatetsky-Shapiro, G.: Discovery, analysis and presentation of strong rules. In: Knowledge Discovery in Databases, pp. 229–248 (1991)
16. Srikant, R., Agrawal, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the International Conference on Very Large Databases, pp. 478–499 (1994)
17. Yan, X., Zhang, C., Zhang, S.: Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. Expert Systems with Applications: An International Journal 36(2), 3066–3076 (2009)
18. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm. In: EUROGEN, vol. 3242(103), pp. 95–100 (2001)
19. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)

Supervised Rule Based Thermodynamic Cycles Design Technique

Ramon Ferreiro Garcia[1] and Jose Luis Calvo Rolle[2]

[1] ETSNM, Dept. Industrial Eng. University of La Coruna
ferreiro@udc.es
[2] EUP, Dept. Industrial Eng. University of La Coruna
jlcalvo@udc.es

Abstract. Most of process design tasks can be carried out by defining a sequence of steps into which actions under some conditions are to be performed. In this way the design of complex or combined thermodynamic cycles to be applied on industrial thermal plants match such a design patter for which hybrid intelligent techniques are being used. In this work a Sequential Function Chart driven design methodology implemented as a supervised rule based system is proposed and applied on thermodynamic complex cycle design tasks. The applied strategy provides us an agile and deterministic thermal cycle design methodology.

Keywords: Database, Expert systems, Petri nets, Sequential function chart, Rule based system.

1 Introduction

Process industry generally and specifically chemical engineering process design and exploitation techniques requires sophisticated tools or software based facilities capable to provide us the way to increase efficiency at design and production levels.

Many complex problems inherent to the process industry including design and implementation planning strategies, require hybrid intelligent systems techniques that integrate several intelligent techniques including expert or rule based systems, fuzzy logic, neural networks, neuro-fuzzy and learning algorithms including genetic algorithms [1],[2],[3] and [4]. Some of this techniques are being applied to achieve the proposed abjectives.

The aim of this work is to develop a soft based tool to be applied on the research task of finding the most convenient state point characteristics of thermodynamic cycles under selected working fluids using data of the available NIST database. With independence of the methodology applied to develop the mentioned tool, a supervised expert system capable for providing a deterministic design guide is required. Such requirements are fulfilled with the help of sequential function charts design methodology.

A Sequential function chart (SFC) is a graphical programming language used for industrial applications. It is one of the five languages defined by IEC 61131-3

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 327–335, 2011.

standard, being defined in IEC 848, "Preparation of function charts for control systems", and is an evolution of the binary Petri Nets theory. (A Petri Net is a graphical tool for the description and analysis of concurrent processes which arise in systems with many components such as in distributed systems), [5], [6] and [7]. It is commonly used to program sequential hybrid processes that can be split into steps.

The main components of a SFC include:

- Steps with associated actions
- Transitions with associated logic conditions
- Directed links between steps and transitions

Steps contain actions and transitions contain Boolean conditions.

Steps in an SFC diagram can be active or inactive. Actions are only executed for active steps. A step can be activated for one of two motives:

(a), it is an initial step as specified by the programmer.

(b), it was activated during a scan cycle by the previous transition and not deactivated by the next transition with associated conditions.

Steps are activated when all steps above it are active and the connecting transition is associated is true. When a transition is passed, all steps above are deactivated at once and after all steps below are activated at once.

SFC is an inherently parallel language in that multiple control flows can be active. Specifically to this work, the developed SFC driven expert system consists of a supervised sequential process which satisfies the data demands required for an expert system destined to define the multidimensional space of a thermodynamic cycle driven thermal process analysis.

Nevertheless, SFC based applications can be implemented by means of any structured text based program such as for instance Fortran or C source code. Consequently the developing methodology consists inherently in a supervised rule based system to efficiently manage a database as well as the achieved results.

The SFC can be described by means of a causal sequential flow graph. Various classical SFC design methods have been previously reported during the last four decades. Nevertheless, focused on successful industry applications, recently appeared some classic books on the subject with relevant applications concerning our developments [8], [9], [10], [11].

2 The Context of the Problem

Many data driven processes can be represented, described and solved by means of sequential function charts or at least described by causal signal graphs. The problem of access a database to retrieve the corresponding data following the proper user specifications to compute a set of functions with recurrence capabilities to further compute a complete search space, can be solved sequentially [12], [13] and [14].

A Rule based system processed according a scheduler is applied. The core of the required scheduler is a SFC. Proposed SFC obeys to a set of actions executed under certain associated conditions to be sequentially executed. The fact of processing deterministically the set of actions under its associated conditions taking into account

that no action can be performed before a necessary condition is met, implies the category of a supervised hybrid sequential system.

A block diagram of the applied search strategy is shown in figure 1. The proposed strategy has the following structure:

(a) A module to enter external data where the user defines the thermodynamic cycle structure, for which the appropriated and specific data must be entered from a well structured and consistent database [15]. The user selects the cycle type and provides the necessary information.

(b) A module where a rule based system computes the selected cycle providing enthalpies to further compute the cycle efficiency.

(c) A Recurrent loop is applied returning to repeat the past computations the amount of times necessary to map the efficiency as function of operating variables in order to have the necessary information to select the most convenient cycle according some economic criteria.

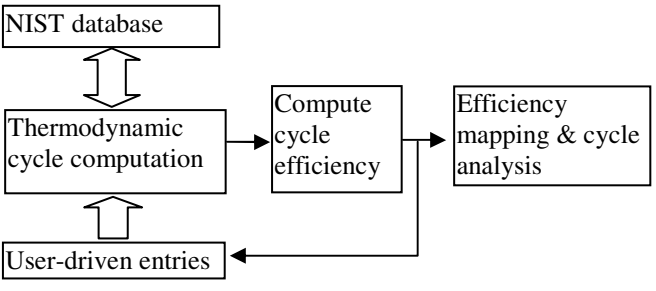


Fig. 1. Block diagram of the design methodology

3 Problem Solving Task

The core of the task is synthesized by the data flow diagram shown in figure 2. A set of initial data must be selected according the type of cycle, working fluid and efficiency criteria.

Subsequently, the first step of the loop is executed retrieving a set of enthalpies necessary to compute the associated efficiency of the cycle under actual conditions. This sequence of actions must be repeated till the while loop finish, where for every loop stage a different set of values for the pressure and temperature (P(i),T(i)) is applied, where (i) is the ith state point of the considered cycle. The efficiency space of the selected thermodynamic cycle is then mapped. Representation of the searched efficiency space is then carried out by some functional approximation technique in order to highlight the relevant characteristics of the studied cycle. In this particular case relevant information is achieved with respect to the efficiency as a result of changing the two variables (pressure and temperature) of the considered state point of the selected thermodynamic cycle. In this way, if the high and low temperatures (T(H), T(L)) as well as high and low pressures (P(H), P(L)) are to be studied to determine its influence on the efficiency, then the efficiency of the considered cycle

could be represented by means of a feedforward neural network with four input variables and an output variable, as shown in figure 3. Such consideration obeys to the fact that the efficiency is a function of the considered thermal cycle variables according the expressions (1) and (2).

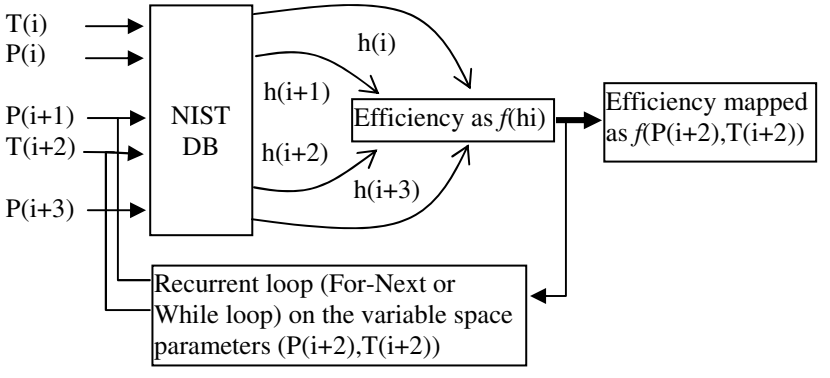


Fig. 2. Data flow diagram of the implemented

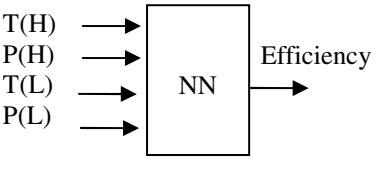


Fig. 3. Representation of the efficiency as function of four variables

$$h(i,j) = f(T(i,j), P(i,j), S(i,j)) \cdot \tag{1}$$

$$eff = f(h(i,j)) \cdot \tag{2}$$

4 A Case Study

The studied case consists of a basic Rankine Cycle with four state points where the n-dimensional efficiency space is selected as two degrees of freedom (2-DOF), where the input variables are the pressure and temperature and the output is the efficiency.

Database access dynamics can be represented by a sequence of data transfer events such as it is depicted in figure 4. Taking into account that P(i), T(i), S(i) and h(i) are respectively the pressure, temperature, entropy and enthalpy of the considered working fluid for each considered state point (i), in figure 3 it is shown the sequence of data access actions (application oriented) executed to retrieve the required data

from the database NIST, and synthesize a three-dimensional searching space. Such thermodynamic efficiency space achieved in this case for the thermal cycle efficiency as function of the pressure and temperature of the state point 4, allows us to have an insight of the problem space so that decision making (DM) results for further cycle analysis is a trivial task.

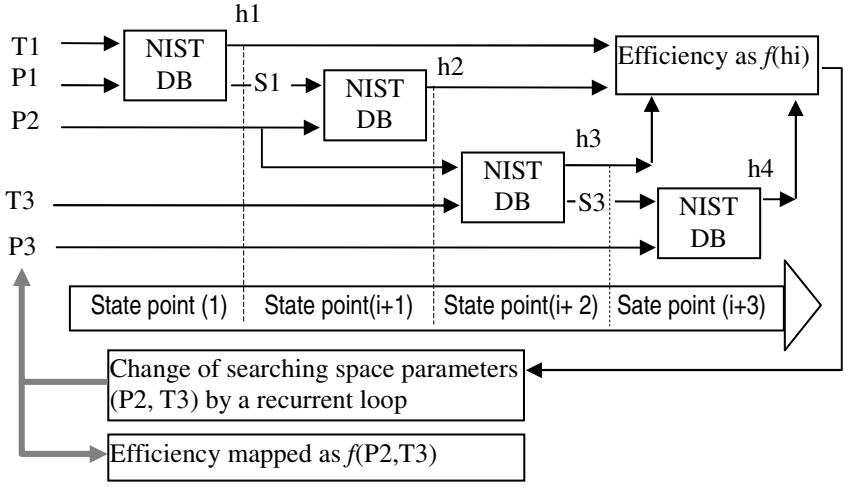


Fig. 4. Sequential data flow diagram representing a sequence of data transfer events

The SFC shown in figure 4 is an application oriented task consisting of the sequence of sub-tasks necessary to achieve the n-dimensional representation of the thermal efficiency into the search space. Implementation can be carried out by means of any commercial specific software application such as the software used in PLC based automation applications. The structured text programming language of the standard IEC-61131-3 is a powerful facility. Retrieved data from the database after processing the task described by the SFC depicted in figure 5 is shown in tables 1 to 5. In above tables, only the useful data is achieved since the rest of data into the database is ignored for our purposes.

Table 1. The searched efficiency space as function of the temperature and pressure for the first space scan

State point	T(K)	P(bar)	h (kJ/kg)	S (kJ/kg-K)
1	330	0.20	238	0.79
2	331	230	261	0.79
3	670	230	2659	5.28
4	333	0.20	1733	5.28

Table 2. The searched efficiency space as function of the temperature and pressure for the second space scan

State point	T(K)	P(bar)	h (kJ/kg)	S (kJ/kg-K)
1	330	0.20	238	0.79
2	332	250	261	0.79
3	690	250	2759	5.58
4	340	0.20	1753	5.58

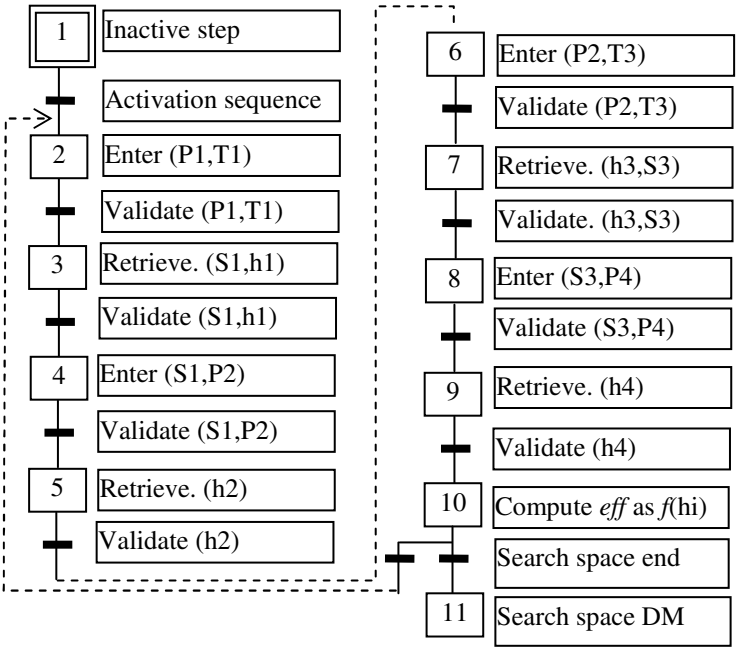


Fig. 5. SFC based scheme to solve the supervised data flow problem

Table 3. The searched efficiency space as function of the temperature and pressure for the third space scan

State point	T(K)	P(bar)	h (kJ/kg)	S (kJ/kg-K)
1	330	0.20	238	0.79
2	333	270	261	0.79
3	710	270	2859	5.80
4	363	0.20	1793	5.80

Table 4. The searched efficiency space as function of the temperature and pressure for the fourth space scan

State point	T(K)	P(bar)	h (kJ/kg)	S (kJ/kg-K)
1	330	0.20	238	0.79
2	334	300	261	0.79
3	730	300	2959	6.18
4	333	0.20	1833	6.18

Table 5. The searched efficiency space as function of the temperature and pressure for the fifth space scan

State point	T(K)	P(bar)	h (kJ/kg)	S (kJ/kg-K)
1	330	0.20	238	0.79
2	335	230	261	0.79
3	770	340	2659	6.25
4	333	0.20	1873	6.25

The information of above tables is now used to determine the thermal efficiency as function of the selected space parameters (P3, T3) for this application where five recurrent database scan sessions have been performed.

Further information processing requires the application of expressions (3) and (4).

$$h(i, j) = f(T(i, j), P(i, j)) . \tag{3}$$

$$eff(j) = f(h(i, j)) \tag{4}$$

The efficiency space can be represented for a 2-DOF by means of any functional approximation technique such as a conventional feedforward neural network. As consequence of it, the graphical representation is shown in figure 6, where the efficiency is mapped versus the two selected variable parameters of the cycle (T, P). The profile of the picture give us a relevant idea of how the cited variables influence the efficiency in order to take into account the next decisions with regard to cycle improving purposes. Such results have been experimentally verified or contrasted along decades of thermal cycle's implementation, so that no doubt exist about results.

Expression (4) represented by figure 6 have been implemented using functional approximation tools based on feedforward neural networks trained under the backpropagation conjugate gradient algorithm.

The data for NN training task is achieved from [15] database. The feedforward structure of the selected back propagation NN based functional approximation algorithm has been selected for this application as 4-10-1 with acceptable precision. In table 6 there are shown the training characteristics in Matlab code. This structure has been selected after some trial and error tests. The selected training algorithm is the conjugate gradient of Fletcher and Reeves.

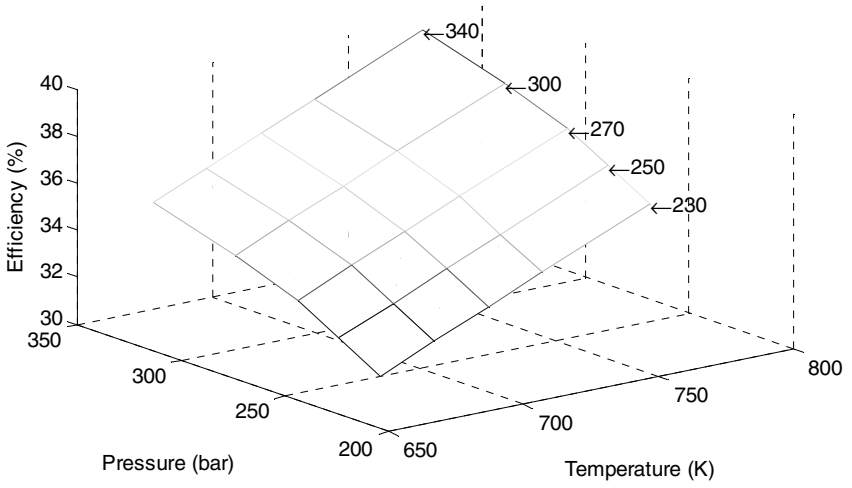


Fig. 6. Thermal cycle efficiency for a 2-DOF search space

Table 6. Neural network training characteristics

Action	Command
Feedforward NN structure and Training algorithm	net=newff(minmax(p),[4,10,1], {'tansig','tansig','purelin'},'traincgf');
Results display	net.trainParam.show = 5;
Training epochs	net.trainParam.epochs=300;
Training command	[net,tr]=train(net,p,t);
NN simulink structure	gensim(net,-1);
Training results	TRAINCGF-srchcha-calcgrad, Epoch 118/300, MSE 1.76e-8/0, Gradient 0.000194/1e-6

5 Conclusions

In this research work, a Sequential Function Chart driven design methodology developed under a supervised rule based system associated to some functional approximation techniques have been proposed, developed and successfully implemented.

The applied modeling technique using functional approximation tools based on feedforward neural networks trained under the backpropagation conjugate gradient algorithm yields satisfactory results providing relevant information available to further analysis. In this way, the following relevant characteristics may be highlighted:

The thermal efficiency is achieved as function of cycle variables of interest to determine further actions.

Based on the results of the feedforward neural networks, it is possible to know the relevance of input variables or its influence on the efficiency.

The facilities to compare several cycles provide us a powerful tool in determining the most convenient thermal cycle.

The analysis carried out on the different WFs provides relevant information which could be useful to optimize the cycles under different optimization criteria.

References

1. 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)
2. Corchado, E., Abraham, A., Carvalho, A.C.P.L.F.D.: Ponce-Leon Ferreira de Carvalho AC: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)
3. 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)
4. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
5. Peterson, J.: Petri Nets, ACM Computing Surveys, vol. 9(3), pp. 223–252 (1977), doi:10.1145/356698.356702
6. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice-Hall, Englewood Cliffs (1981)
7. Reisig, W.: A Primer in Petri Net Design. Springer, Heidelberg (1992)
8. Jensen, K.: Coloured Petri Nets. Springer, Heidelberg (1997)
9. Riemann, R.C.: Modelling of Concurrent Systems: Structural and Semantical Methods in the High Level Petri Net Calculus. Herbert Utz Verlag (1999)
10. Zhou, M., Dicesare, F.: Petri Net Synthesis for Discrete Event Control of Manufacturing Systems. Kluwer Academic Publishers, Dordrecht (1993)
11. Zhou, M., Venkatesh, K.: Modeling, Simulation, & Control of Flexible Manufacturing Systems: A Petri Net Approach. World Scientific Publishing, Singapore (1998)
12. Jackson, P.: Introduction to Expert Systems, Pearson Education Limited, Edinburg Gate, Harlow, Essex CM20 2JE. Addison Wesley Longman Limited, Amsterdam (1999)
13. Giarratano, J.C., Gary, R.: Expert Systems, Principles and Programming. PWS Publishing Company, 20 Park Plaza, Boston, MA 02 116.4324, USA (2005)
14. Darlington, K.: The Essence of Expert Systems. Pearson Education, London (2000)
15. NIST. Reference Fluid Thermodynamic and Transport Properties Database (REFPROP) Version 8.0, U.S. Department of Commerce, Maryland (2007)

Deformation Based Features for Alzheimer's Disease Detection with Linear SVM

Alexandre Savio[1], Manuel Graña[1], and Jorge Villanúa[2]

[1] Grupo de Inteligencia Computacional
www.ehu.es/ccwintco
[2] Osatek, Hospital Donostia Paseo Dr. Beguiristain 109,
20014 San Sebastián, Spain

Abstract. Detection of Alzheimer's disease over brain Magnetic Resonance Imaging (MRI) data is a priority goal in the Neurosciences. In previous works we have studied the accuracy of feature vectors obtained from VBM studies of the MRI data. In this paper we report results working on deformation based features, obtained from the deformation vectors computed by non-linear registration processes. Feature selection is based on the correlation between the scalar values computed from the deformation maps and the control variable. Results with linear kernel SVM reach accuracies comparable to previous best results.

1 Introduction

Alzheimer's disease (AD) is a neurodegenerative disorder, which is one of the most common cause of dementia in old people. Currently, due to the socioeconomic importance of the disease in occidental countries it is one of the most studied. The diagnosis of AD can be done after the exclusion of other forms of dementia but a definitive diagnosis can only be made after a post-mortem study of the brain tissue. This is one of the reasons why early diagnosis based on Magnetic Resonance Imaging (MRI) is a current research hot topic in Neuroscience. One of the current lines of research involves the application of machine learning algorithms to features extracted from brain MRI. We have already explored the application of various machine learning and computational intelligence algorithms to AD prediction [7,15,1,2]. Specifically we have performed these computational experiments on a subset of the Open Access Series of Imaging Studies (OASIS) database [12]. These works involved the use of Voxel-based Morphometry (VBM) [3] to select the Gray Matter (GM) voxels that would serve as discriminant features. We have made public the set of extracted features in order to allow for independent experimentation upon them [8].

Morphometry analysis has become a common tool for computational brain anatomy studies. It allows a comprehensive measurement of structural differences within a group or across groups, not just in specific structures, but throughout the entire brain. In this paper we use Deformation-based Morphometry (DBM) [10,16] and Tensor-based Morphometry (TBM) [5,11] to guide the feature extraction process. These morphometry methods analyze displacement vectors

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 336–343, 2011.

resulting from non-linear registration procedures with high number of degrees of freedom.

A similar study [14] with 50 subjects obtained 92% of accuracy when discriminating AD subjects from healthy controls using features extracted from displacement fields and different classification methods with SVM, Bayes statistics, and voting feature intervals (VFI). In addition, another study [16] obtained 83% of accuracy using similar approaches to detect subjects with mild cognitive impairment. Although their results can not be reproduced, this work confirms that the approach that we follow is a promising area of research.

In this experiment we obtain scalar measures of the voxel displacements and afterwards compute their correlation with the control variable, which indicates if the sample corresponds to a control subject or an AD patient. The voxel sites with high correlation are selected for the extraction of the feature vector values. We report the results of Support Vector Machine (SVM) with linear kernels performing the classification task.

Section Materials and Methods gives a description of the subjects selected for the study, the image processing, feature extraction details and the classifier system. Section Results gives our classification performance results and section Conclusions gives the conclusions of this work and further research suggestions.

2 Materials and Methods

A database of ninety eight women extracted from the freely available OASIS database were used in this AD detection experiment. The demographic and imaging details of the sample can be found elsewhere [8,15]. The implementation of the SVM used for this study is included in the libSVM (`http://www.csie.ntu.edu.tw/ cjlin/libsvm/`) software package and described in detail in [6]. The feature extraction step requires the data to be spatially normalized. The subjects in the database were already linearly registered to a MNI152 template[12]. Taking into account that, we need to non linearly transform them to a common template in order to obtain the deformation fields which will be used as starting point in the feature extraction process. For this non-linear registration step we could have used again the MNI152 standard template, but the registration algorithm used in this study [4] could not cope well with the large deformations required to register some subjects with enlarged ventricles. For this reason, a custom brain template volume was created with all the subjects in the database. This custom template was subsequently non linearly registered to all the study subjects. As a result from the non linear registration, displacement vectors for each subject are obtained. These displacement vector fields describe the effects of deformation of the template brain to the subject's. For each voxel i, the displacement field for one subject have a vector (x_i, y_i, z_i) representing the ending point of voxel i in the registration process.

Two measures have been extracted from the displacement vectors (see Fig. 1):

1. The displacement vector magnitudes, denoted DM in the results section

$$DM_i = \sqrt{x_i^2 + y_i^2 + z_i^2}, \tag{1}$$

2. The Jacobian determinant of the displacement field gradient matrices, denoted JD in the results section

$$\mathbf{J}_i = \begin{pmatrix} \partial(x-u_x)/\partial x & \partial(x-u_x)/\partial y & \partial(x-u_x)/\partial z \\ \partial(y-u_y)/\partial x & \partial(y-u_y)/\partial y & \partial(y-u_y)/\partial z \\ \partial(z-u_z)/\partial x & \partial(z-u_z)/\partial y & \partial(z-u_z)/\partial z \end{pmatrix}. \tag{2}$$

The Jacobian matrix in this case describes the velocity of the deformation procedure in the neighboring area of each voxel. To calculate this matrix, for each voxel site, we used the central difference using two adjacent voxels in one dimension. The determinant of the Jacobian matrix $\mathbf{J}_i$ is commonly used to analyze the distortion necessary to deform the images into agreement. A value $\det(\mathbf{J}_i) > 1$ implies that the neighborhood adjacent to the displacement vector in voxel i was stretched to match the template (i.e., local volumetric expansion), while $\det(\mathbf{J}_i) < 1$ is associated with local shrinkage.

Much of the information about the shape change is lost using these measures, nevertheless a more complex multivariate approach would have to be performed in order to use all the information in the deformation gradient matrices or the Green strain tensors defined as $\mathbf{S}_i = \left(\mathbf{J}_i^T \mathbf{J}_i\right)^{1/2}$ [11]. Once the DM and JD maps were calculated, significant voxels were selected from a correlation measure of the voxels to the class labels of each subject. All the registration procedures in this study were performed using ANTS (http://www.picsl.upenn.edu/ANTS).

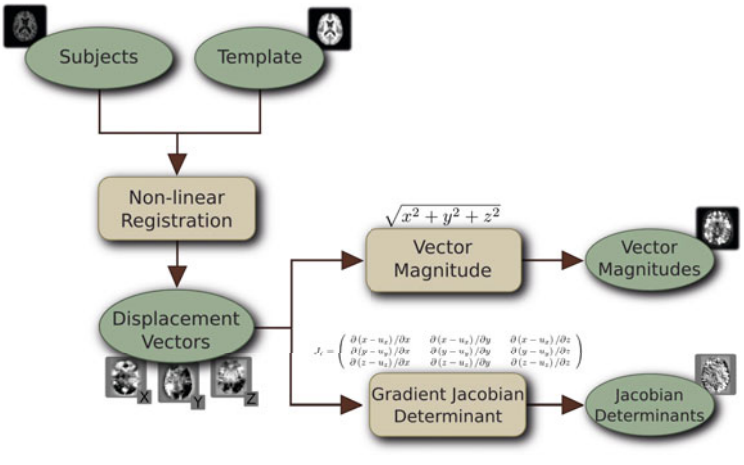


Fig. 1. Pipeline of the image pre-processing steps

The feature values were extracted from the scalar DM and JD maps computed on the displacement vectors resulting from the registration processes, as described in figure 1. For each voxel site i we extracted one vector $\mathbf{v}_i$ with n components being the value of the voxel i of each one of the n subjects of this experiment. Afterwards correlation measures of these vectors with the control variable, specified by the vector containing the subject class label (-1 for control subject or 1 for patient) were performed. Pearson and Spearman correlation were used in this study. Volume masks containing the voxel sites whose correlation values where above some specified percentile (i.e. 0.990, 0.995) for each combination of map and correlation measure were applied to the corresponding DM or JD map to extract the feature vectors. Figure 2 illustrates the process. Features were normalized before training the SVM on them.

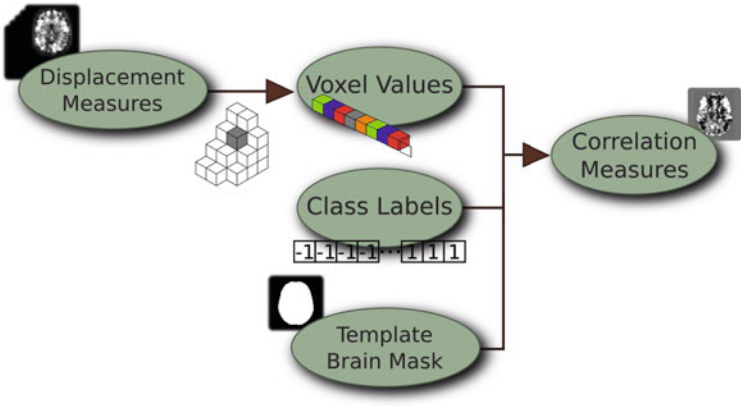


Fig. 2. Pipeline of the calculation of the correlation volumes

2.1 Pearson's Correlation

The Pearson's product-moment correlation coefficient [9] (PMCC or typically denoted by r) is a measure of the correlation (linear dependence) between two variables X and Y, giving a value between $+1$ and -1 inclusive. It is widely used in the sciences as a measure of the strength of linear dependence between two variables. It was developed by Karl Pearson from a similar but slightly different idea introduced by Francis Galton in the 1880s. The correlation coefficient is sometimes called "Pearson's r."

Pearson's correlation coefficient between two variables is defined as the covariance of the two variables divided by the product of their standard deviations:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}, \tag{3}$$

where $\bar{X}$ and$\bar{Y}$ are the mean of the variables X and Y respectively.

2.2 Spearman's Correlation

The Spearman's correlation coefficient is often thought of as being the Pearson's correlation coefficient between the ranked variables. In practice, however, a simpler procedure is normally used to calculate ρ. The n raw scores X_i, Y_i are converted to ranks x_i, y_i and the differences $d_i = x_i - y_i$ between the ranks of each observation on the two variables are calculated.

If there are no tied ranks, then ρ is given by [13]:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{4}$$

If tied ranks exist, Pearson's correlation coefficient between ranks should be used for the calculation 3. One has to assign the same rank to each of the equal values. It is an average of their positions in the ascending order of the values.

3 Results

Classifiers performance was measured with the architecture strategies mentioned before using a 10-fold cross-validation methodology. In this section the following data for each experiment is presented: the number of features extracted from each subject, classification accuracy, sensitivity, which is related to AD patients and specificity, which is related to control subjects. The results shown are the mean and standard deviation (stdev) values of the classification results from the cross-validation process. In this paper we report the results of the linear kernel SVM. Each of the tables contain the results of using all the selected voxel sites with correlation value above the selected percentile (top row), and the results selecting a fixed number of voxel sites in descending order of correlation value. The objective is to see if the feature vector size reduction based only on the correlation magnitude is an efficient feature selection method.

The results of the features extracted on the basis of the Spearman's correlation are systematically worse the results obtained by the SVM on the features selected from the VBM analysis [15]. These results are in tables 1, 2 and 3. Results performing the feature extraction on the DM map with the 0.995 percentile (table 1) improve over the 0.999 percentile (table 2), reaching values close to the reference values. The application of the process to the JD map (table 3) does not improve the results.

The results of the feature vectors extracted on the basis of the Pearson's correlation improve on the results of the Spearman's correlation selection. These results are in tables 4 and 5. The results on the DM map (table 4) are comparable to the reference results in [15]. Note that the use of the full feature vector give the same result as the reduced vector of 250 voxel sites, suggesting that a strong feature vector size reduction can be achieved. Results on the JD map (table 5) are worse than the DM results.

Table 1. Results using linear SVM on DM features obtained from the 0.995 percentile of the Spearman correlation

#Features	Accuracy	Sensitivity	Specificity
12229	0.76 (0.15)	0.77 (0.28)	0.75 (0.17)
2000	0.79 (0.14)	0.82 (0.12)	0.75 (0.20)
1000	0.79 (0.10)	0.87 (0.13)	0.70 (0.16)
500	0.77 (0.13)	0.87 (0.17)	0.67 (0.20)
250	0.79 (0.10)	0.90 (0.13)	0.67 (0.17)

Table 2. Results using linear SVM on DM features obtained from the 0.999 percentile of the Spearman correlation

#Features	Accuracy	Sensitivity	Specificity
1861	0.66 (0.14)	0.70 (0.20)	0.62 (0.21)
1000	0.66 (0.14)	0.77 (0.18)	0.55 (0.20)
500	0.71 (0.12)	0.85 (0.17)	0.57 (0.17)
250	0.72 (0.15)	0.80 (0.11)	0.65 (0.29)

Table 3. Results using linear SVM on normalized JD features obtained from the 0.995 percentile of the Spearman correlation measures

#Features	Accuracy	Sensitivity	Specificity
17982	0.76(0.14)	0.77(0.27)	0.75(0.16)
2000	0.65(0.15)	0.65(0.21)	0.65(0.24)
1000	0.60(0.16)	0.62(0.27)	0.57(0.31)
500	0.58(0.08)	0.70(0.20)	0.47(0.18)
250	0.61(0.09)	0.65(0.21)	0.57(0.20)

Table 4. Results using linear SVM on DM features over the 0.995 percentile of the Pearson correlation measures

#Features	Accuracy	Sensitivity	Specificity
27474	0.84 (0.10)	0.90 (0.17)	0.77 (0.14)
2000	0.79 (0.12)	0.85 (0.17)	0.72 (0.08)
1000	0.79 (0.10)	0.90 (0.13)	0.67 (0.17)
500	0.79 (0.13)	0.85 (0.21)	0.72 (0.14)
250	0.84 (0.10)	0.92 (0.12)	0.75 (0.17)

Table 5. Results using linear SVM on JD features over the 0.990 percentile of the Pearson correlation measures

#Features	Accuracy	Sensitivity	Specificity
43967	0.66 (0.19)	0.70 (0.20)	0.62 (0.24)
2000	0.75 (0.13)	0.75 (0.26)	0.75 (0.17)
1000	0.69 (0.15)	0.77 (0.14)	0.60 (0.24)
500	0.66 (0.16)	0.72 (0.30)	0.60 (0.17)
250	0.66 (0.17)	0.70 (0.20)	0.62 (0.24)

4 Conclusions

In this paper we report classification results on the application of a feature extraction process based on the deformation vectors obtained from non-linear registration processes. The sample is a subset of the OASIS database carefully selected to be pairwise comparable. From the displacement vectors we computed two scalar measures: the magnitude of the displacement vector and the Jacobian determinant of the displacement gradient matrix. We compute the Spearman's and Pearson's correlations of the voxel values with the control variable, selecting voxel sites with the higher correlation. Results show that the deformation vector magnitude features provide better classification accuracy reaching the values of the reference results. Results for the Pearson features are better than for the Spearman features. Although we did not use any combination of classifiers in this study [17], it is important to note how the fusion of information from different images was used to extract relevant features for the classification task. This process of fusion of images of different modalities can be found in a wide variety of medical imaging studies. We are working on the application of non-linear SVM, using RBF kernels, and extending the experimental exploration to other classifiers and combinations.

Acknowledgments

We thank the Washington University ADRC for making MRI data available.

References

1. Savio, A., García-Sebastián, M., Hernández, C., Graña, M., Villanúa, J.: Classification results of artificial neural networks for alzheimer's disease detection. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 641–648. Springer, Heidelberg (2009)
2. Savio, A., García-Sebastián, M., Graña, M., Villanúa, J.: Results of an Adaboost Approach on Alzheimer's Disease Detection on MRI. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5602, pp. 114–123. Springer, Heidelberg (2009)
3. Ashburner, J., Friston, K.J.: Voxel-Based Morphometry: The Methods. Neuroimage 11(6), 805–821 (2000)
4. Avants, B.B., Epstein, C.L., Grossman, M., Gee, J.C.: Symmetric diffeomorphic image registration with Cross-Correlation: evaluating automated labeling of elderly and neurodegenerative brain. Medical image analysis 12(1), 26–41 (2008)
5. Bossa, M., Zacur, E., Olmos, S.: Tensor-based morphometry with stationary velocity field diffeomorphic registration: Application to ADNI. NeuroImage 51(3), 956–969 (2010)
6. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines, Software (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
7. Chyzhyk, D., Graña, M., Savio, A., Maiora, J.: Hybrid Dendritic Computing with Kernel-LICA applied to Alzheimer's disease detection in MRI. Neurocomputing, (2011) (accepted)

8. Chyzhyk, D., Savio, A.: Feature extraction from structural MRI images based on VBM: data from OASIS database. Technical Report GIC-UPV-EHU-RR-2010-10-14, Grupo de Inteligencia Computacional UPV/EHU (2010)
9. Cohen, J.: Statistical Power Analysis for the Behavioral Sciences. In: Routledge Academic, 2nd edn. (1988)
10. Gerardin, E., Chetelat, G., Chupin, M., Cuingnet, R., Desgranges, B., Kim, H.-S., Niethammer, M., Dubois, B., Lehericy, S., Garnero, L., Eustache, F., Colliot, O.: Multidimensional classification of hippocampal shape features discriminates Alzheimer's disease and mild cognitive impairment from normal aging. NeuroImage 47(4), 1476–1486 (2009)
11. Lepore, N., Brun, C., Chou, Y.Y., Chiang, M.C., Dutton, R.A., Hayashi, K.M., Luders, E., Lopez, O.L., Aizenstein, H.J., Toga, A.W., Becker, J.T., Thompson, P.M.: Generalized tensor-based morphometry of HIV/AIDS using multivariate statistics on deformation tensors. IEEE Transactions on Medical Imaging 27(1), 129–141 (2008)
12. Marcus, D.S., Wang, T.H., Parker, J., Csernansky, J.G., Morris, J.C., Buckner, R.L.: Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. Journal of Cognitive Neuroscience 19(9), 1498–1507 (2007)
13. Maritz, J.S.: Distribution-Free Statistical Methods, 2nd edn. Chapman and Hall, Boca Raton (April 1995)
14. Plant, C., Teipel, S.J., Oswald, A., Böhm, C., Meindl, T., Mourao-Miranda, J., Bokde, A.W., Hampel, H., Ewers, M.: Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease. NeuroImage 50(1), 162–174 (2010)
15. Savio, A., García-Sebastián, M., Chyzhyk, D., Hernández, C., Graña, M., Sistiaga, A., Lopez de Munain, A., Villanúa, J.: Neurocognitive disorder detection based on feature vectors extracted from VBM analysis of structural MRI. Computers in Biology and Medicine (2011) (accepted with revisions)
16. Teipel, S.J., Born, C., Ewers, M., Bokde, A.L.W., Reiser, M.F., Möller, H.-J., Hampel, H.: Multivariate deformation-based analysis of brain atrophy to predict Alzheimer's disease in mild cognitive impairment. NeuroImage 38(1), 13–24 (2007)
17. 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 System for Survival Analysis after EVAR Treatment of AAA

Josu Maiora and Manuel Graña

Computational Intelligence Group, University of the Basque Country

Abstract. Abdominal Aortic Aneurysm (AAA) is a local dilation of the Aorta that occurs between the renal and iliac arteries. Recently, the procedure used for treatment involves the insertion of a endovascular prosthetic (EVAR), which has the advantage of being a minimally invasive procedure but also requires monitoring to analyze postoperative patient outcomes. The most widespread method for monitoring is the computerized axial tomography (CAT) imaging, from which we can make 3D reconstructions and segmentations of the aorta (lumen) of the patient under study. Based on a previously published method to measure the deformation of the aorta between two studies of the same patient using registration techniques, in this paper we apply neural network classifiers to the registration results to build a predictor of the patient survival. This would provide an additional tool for decision support to the medical team.

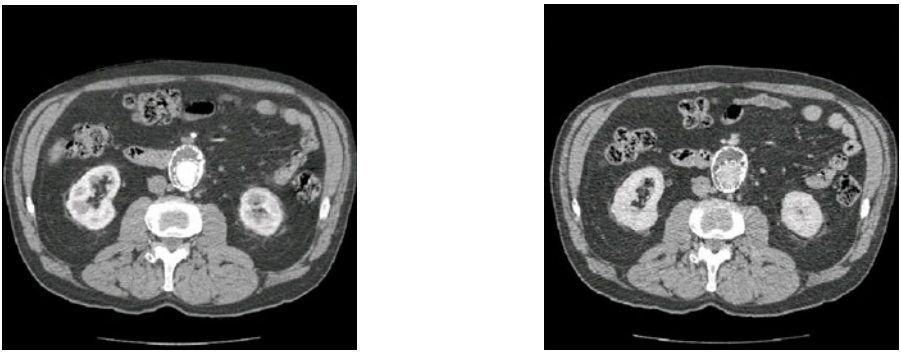
Keywords: Medical Image, Registration, Neural Networks.

1 Introduction

Decision suport systems have a growing importance in the clinical routine and artificial neural networks [5,6,13] provide an important tool in the development of computational decision support systems. Their use has been described for a number of medical classification tasks[8]. Among the desirable properties of artificial neural networks is their ability to perform complex pattern-recognition tasks and the fact that they do not require foreknowledge of diagnostic rules. In this paper, we present a hybrid[2,1,14,4] survival prediction system based on artificial neural networks and image registration methods. Specifically, the input to the artificial neural network is the result of the registration methods. The problem of Abdominal Aortic Aneurysms (AAA) and its treatment Endovascular Aneurysm Repair (EVAR) has been presented already in [10,9]. The most widely used technique for EVAR monitoring is to obtain Computerized Tomography (CT) images of the abdominal region after an intravenous contrast agent has been injected (Fig. 1).

Such scans of the patient's abdominal area are available in the clinical routine as a set of 2D images whose visual analysis is time-consuming. The aim of our work is to make an automatic analysis of the AAA using digital image processing techniques, yield visual and quantitative information for monitoring

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 344–351, 2011.



(a) (b)

Fig. 1. Comparison of vessel intensity values between CT and CTA slice. a)CTA slice using the contrast agents, blood in lumen is highlighted for a better view. b)CT slice without using the contrast agent, intensity values of lumen and thrombus are similar.

and tracking of patients who underwent EVAR and classify them as favorable or unfavorable. In our approach, first we estimate the rigid motion of the lumen relative to the spinal cord [11] as well as its deformation [12,3]. Visual overlapping of such transformed data can help identifying deformation patterns having a high probability of dangerous progression of the aneurysm but, we need to quantify those data to use neural network classifiers. The aim of our research is to calculate the similarity metrics after rigid, affine and deformable registration of the aortic lumen after EVAR and use these variables to construct a neural network to make a prediction about future complications and disease progression.

The final goal of this paper was to test the hypothesis that classification algorithms constructed using Backpropagation Neural Network approach can discriminate between favorable and unfavorable evolution of patients who underwent an EVAR procedure.

2 Methods

First, the lumen is segmented using a 3D region growing algorithm. After that, the registration of the lumen extracted from two datasets of the same patient obtained at different moments in time is computed and then, we quantify the deformations of the lumen computing the similarity metrics between the reference and the registered dataset after different registrations modalities: rigid, affine and deformable. Finaly we classify them as favoralble or unfavoralble using a neural netork. Next, we proceed to describe each component of the system.

2.1 Region Growing Based Lumen Segmentation

Images obtained by Computerized Tomography (CT) are visualized as 3D volumes with appropriate software tools[7]. Then, a segmentation process of lumen

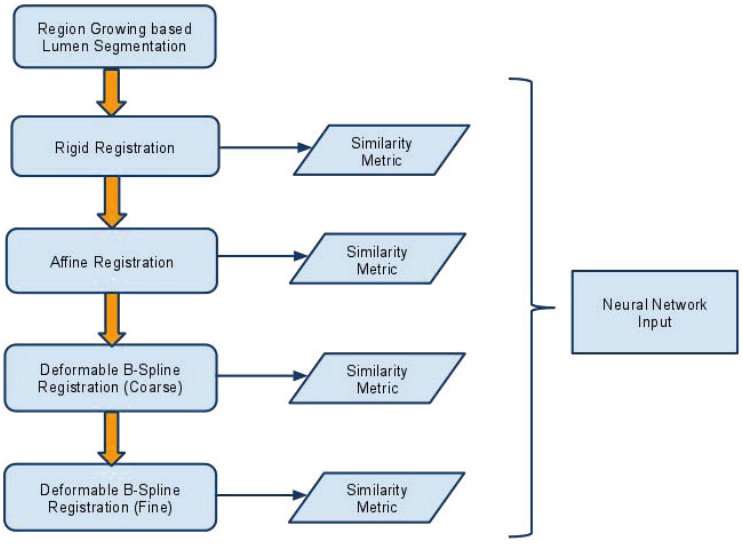


Fig. 2. Pipeline of the neural network input generation process

is carried out. We have used a User-Guided Level Set Segmentation (UGLSS) [16] based on the well-known 3D active contour segmentation method called Region Competition [17]to get segmented image of the lumen. The UGLSS algorithm proceeds as follows: First, the image is resampled into a volume with isotropic spacing (1,1,1). Then, a Region of Interest (ROI) that contains the structure to be segmented, the lumen, is selected. During the preprocessing, probability maps are computed, by applying a smooth lower and upper threshold. This ensures that voxels inside the lumen have a positive value and the outside negative. We place a seed to initialize the evolving contour into the lumen and we establish the parameters that control the propagation velocity and curvature velocity. An evolving contour is a closed surface C(t,u,v) parameterized by variables u, v and by the time variable t. The contour evolves according to the following partial differential equation (PDE):

$$\frac{\partial}{\partial t}C(t,u,v) = F\overrightarrow{N} \tag{1}$$

We compute the external force F by estimating the probability that a voxel belongs to the structure of interest and the probability that it belongs to the background at each voxel in the input image:

$$F = \alpha(P_{obj} - P_{bg}) + \beta_K \tag{2}$$

2.2 Registration

A sequence of three registration steps is performed; rigid, affine and deformable (B-splines) registrations. The first study is considered the fix image and the

others are registered respect to it. A linear interpolator, Mutual Information metric and Regular Step Gradient Descent Optimizer are used. Rigid, affine and deformable registration of the lumen allows for visual comparison of the evolution of the stent-graft. First the two images corresponding to the patient lumen are roughly aligned by using a transform initialization and then the two images are registered using a rigid transformation. The rigid transformation is used to initialize a registration with an affine transform of the stent graft. The transform resulting from the affine registration is used as the bulk transform of a BSpline deformable transform and finally the deformable registration is computed.

We use two similarity metrics: the sum of squared intensity differences (SSD) and mutual information (MI). These similarity metrics have each been used widely in the past for nonrigid registration, to measure the intensity agreement between a deforming image and the target image. We briefly describe both distances in this section[15].

SSD is suitable when the images have been acquired through similar sensors and thus are expected to present the same intensity range and distribution. For voxel locations x_A in image A, within an overlap domain $\Omega^T_{A,B}$, comprising N voxels:

$$SSD = \frac{1}{N} \sum_{x_A \in \Omega^T_{A,B}} |A(x_A) - B^T(x_A)|^2 \tag{3}$$

Mutual information is a measure of how much information one random variable has about another. The information contributed by the images is simply the entropy of the portion of the image that overlaps with the other image volume, and the mutual information is a measure of the joint entropy respect to the marginal entropies.

$$I(A, B) = H(A) + H(B) - H(A, B) \tag{4}$$

where I(A,B) is the mutual information, H(A) and H(B) are the marginal entropies of the fixed and moving images, and H(A,B) is the joint entropy. We have computed the mean squares and mutual information similarity metrics for the evaluation of the registration in 3 registration processes, each of them with rigid, affine, deformable coarse and deformable fine methods . A decrease of both metric is observed in the consequent registration methods.

2.3 Neural Network Classification Algorithms

We deal with two class classification problem, given a collection of training/ testing input feature vectors $X = \{\mathbf{x}_i \in \mathbb{R}^n,\ i = 1, \ldots, l\}$ and the corresponding labels $\{y_i \in \{-1, 1\},\ i = 1, \ldots, l\}$, which sometimes can be better denoted in aggregated form as a binary vector $\mathbf{y} \in \{-1, 1\}^l$.

Backward propagation of errors, or backpropagation (BP), [13]is a non-linear generalization of the squared error gradient descent learning rule for updating the weights of artificial neurons in a single-layer perceptron, generalized to feed-forward networks, also called Multi-Layer Perceptron (MLP). Backpropagation requires that the activation function used by the artificial neurons (or "nodes") is differentiable with its derivative being a simple function of itself. The backpropagation of the error allows to compute the gradient of the error function relative to the hidden units. It is analytically derived using the chain rule of calculus. During on-line learning, the weights of the network are updated at each input data item presentation. We have used the Levenberg-Marquardt backpropagation optimization.

We restrict our presentation of BP to train the weights of the MLP for the current two class problem. Let the instantaneous error E_p be defined as:

$$E_p(\mathbf{w}) = \frac{1}{2}\left(y_p - z_K(\mathbf{x}_p)\right)^2, \tag{5}$$

where y_p is the p-th desired output y_p, and $z_K(x_p)$ is the network output when the p-th training exemplar x_p is inputted to the MLP composed of K layers, whose weights are aggregated in the vector $\mathbf{w}$. The output of the j-th node in layer k is given by:

$$z_{k,j}(\mathbf{x}_p) = f\left(\sum_{i=0}^{N_{k-1}} w_{k,j,i} z_{k-1,i}(\mathbf{x}_p)\right), \tag{6}$$

where $z_{k,j}$ is the output of node j in layer k, N_k is the number of nodes in layer k, $w_{k,j,i}$ is the weight which connects the i-th node in layer $k-1$ to the j-th node in layer k, and f (ï¿œ) is the sigmoid nonlinear function, which has a simple derivative:

$$f'(\alpha) = \frac{df(\alpha)}{d\alpha} = f(\alpha)(1 - f(\alpha)). \tag{7}$$

The Levenberg-Marquardt algorithm is a very simple, but robust, method for approximating a function. It consists in solving the equation:

$$(J^T J + \lambda I)\delta = J^T E \tag{8}$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and E is the error vector containing the output errors for each input vector used on training the network. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix. λ is the Levenberg's damping factor and δ is the weight update vector that we want to find. The δ tell us by how much we should change our network weights to achieve a (possibly) better solution. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix can be approximated as $H = J^T J$ and the gradient can be computed as $g = J^T E$.

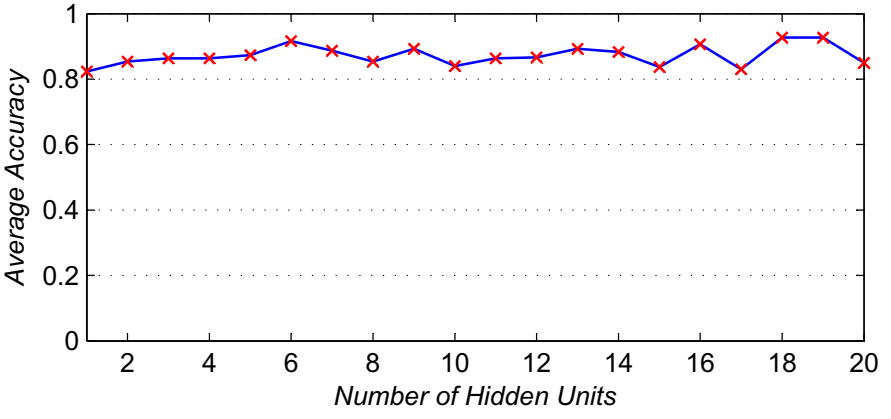


Fig. 3. Average Accuracy for Backpropagation Neural Network varying the number of hidden layers from 1 to 20

3 Results

We have tested the approach with 8 datasets corresponding to 2 patients which have been treated with stent-graft devices. Four datasets are validated by the doctors as having a favorable evolution - first patient- and the other four as unfavorable -second patient- . The CT image stacks consists of datasets obtained from a LightSpeed16 CT scanner (GE Medical Systems, Fairfield, CT, USA) with 512x512x354 voxel resolution and 0.725x0.725x0.8 mm. spatial resolution. The time elapsed between different studies of the same subject varies between 6 and 12 months. We have computed the mean squares and mutual information similarity metrics for the evaluation of the registration. A decrease of both metric is observed in the consecutive registration methods. [9].

We build the feature vectors with the values of the different registration modalities for each dataset. So, we have 8 features for each registered image pair. We use units with sigmoid transfer function trained on the mean squared error performance function and Levenberg-Marquardt optimization with a minimum performance gradient of 1e-10.

We train over the set of features different neuronal architectures varying the number of neurons in the hidden layers from 1 to 20. We train the network 50 times for each number of hidden units and we obtain the average accuracy. An increase in the performance is observed from 1 to 6 hidden units, then the obtained results are variable and we can even notice that due to the Hughes effect, for a large number of hidden layers the performace does not improve becouse of overfitting. The results for leave-one-out cross-validation are represented in Fig. 3.

4 Conclusion and Future Work

After segmentation of the lumen, a registration process is carried out over binary images improving on the works that perform registration over point sets, which always involve a greater loss of information. Registering images from different datasets from a given patient can provide us quantified values of deformation of the stent graft.

The feature vectors have been built with the similarity measures of the segmented lumen after rigid, affine and deformable registration. The datasets of the patients have been previously validated by the medical team as having a favolable or unfavorable evolution.

Therefore, considering the average accuracy data obteined, our main conclusion is that the proposed feature extraction is very effective in providing a good discrimination between patients that can easily be exploited by the classifier construction algorithms and could lead to a model that would predict the evolution of other patients and provide support for the decision. Further ongoing works with a more extensive database is on the way to confirm our conclusions in the framework of collaboration with a team of medical clinical experts. It would also be necessary to compare the results with other binary classifiers as Support Vector Machines (SVM), Relevance Vector Machine, RVM, Logistic Model Tree, LMT, Adaboost as a boosting method, or radial basis functions Neural Networks, RBF.

References

1. Abraham, A., Corchado, E., Corchado, J.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
2. Corchado, E., Abraham, A., Ponce, A.: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)
3. Demirci, S., Manstad-Hulaas, F., Navab, N.: Quantification of abdominal aortic deformation after evar. In: SPIE Medical Imaging, Orlando, Florida, USA (February 2009)
4. Derrac, J., Garcia, S., Herrera, F.: A first study on the use of coevolutionary algorithms for instance and feature selection. Hybrid Artificial Intelligence Systems, 557–564 (2010)
5. Hagan, M., Demuth, H., Beale, M., et al.: Neural network design, PWS Boston, MA (1996)
6. Haykin, S., Network, N.: A comprehensive foundation. Neural Networks (1999)
7. Ibanez, L., Schroeder, W., Ng, L., Cates, J., et al.: The itk software guide (2003)
8. Lim, C., Harrison, R., Kennedy, R.: Application of autonomous neural network systems to medical pattern classification tasks. Artificial intelligence in medicine 11(3), 215 (1997)
9. Maiora, J., García, G., Macía, I., Legarreta, J., Boto, F., Paloc, C., Graña, M., Abuín, J.: Thrombus volume change visualization after endovascular abdominal aortic aneurysm repair. Hybrid Artificial Intelligence Systems, 524–531 (2010)
10. Maiora, J., Garcia, G., Tapia, A., Macia, I., Legarreta, J.H., Paloc, C., De Blas, M., Grana, M.: Thrombus change detection after endovascular abdominal aortic aneurysm repair. International Journal of Computer Assisted Radiology and Surgery 5 (Suppl. 1), S15 (2010)

11. Mattes, J., Steingruber, I., Netzer, M., Fritscher, K., Kopf, H., Jaschke, W., Schubert, R.: Spatio-temporal changes and migration of stent grafts after endovascular aortic aneurysm repair. In: Lemke, H.U., Inamura, K., Doi, K., Vannier, M.W., Farman, A.G. (eds.) Computer Assisted Radiology and Surgery CARS 2005, vol. 1281, pp. 393–397 (2005)
12. Mattes, J., Steingruber, I., Netzer, M., Fritscher, K., Kopf, H., Jaschke, W., Schubert, R.: Quantification of the migration and deformation of abdominal aortic aneurysm stent grafts. SPIE 6144, 61440V (2006)
13. Rumelhart, D., Hinton, G.: learning representations by backpropagating errors. Nature, 533–536 (1986)
14. Wozniak, M., Zmyslony, M.: Designing fusers on the basis of discriminants–evolutionary and neural methods of training. Hybrid Artificial Intelligence Systems, 590–597 (2010)
15. Yanovsky, I., Leow, A.D., Lee, S., Osher, S.J., Thompson, P.M.: Comparing registration methods for mapping brain change using tensor-based morphometry. Medical Image Analysis 13(5), 679–700 (2009)
16. Yushkevich, P.A., Piven, J., Hazlett, H.C., Smith, R.G., Ho, S., Gee, J.C., Gerig, G.: User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. Neuroimage 31(3), 1116–1128 (2006)
17. Zhu, S., Yuille, A.: Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. IEEE Transactions On Pattern Analysis And Machine Intelligence 18(9), 884–900 (1996)

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

José Luis Calvo-Rolle[1], Emilio Corchado[2], Amer Laham[2], and Ramón Ferreiro García[1]

[1] Department de Ingeniería Industrial, Universidad de La Coruña
Avda. 19 de febrero, s/n, 15405, Ferrol, A Coruña, Spain
jlcalvo@cdf.udc.es, ferreiro@udc.es
[2] Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
escorchado@usal.es, amerlaham@usal.es

Abstract. This study presents a novel hybrid generic decision method to obtain the best parameters of a PID (Proportional-Integral-Derivative) controller for desired specifications. The method used is to develop a ruled-based conceptual model of knowledge based system for PID design in Open-Loop. The study shows a hybrid system based on the organization of existing rules and a new way to obtain other specific ones based on decision trees. The model achieved chooses the best controller parameters, between different open loop tuning methods. For this purpose an automatic classification of a huge dataset is used. Data was obtained by applying considered tuning methods to a collection of representative systems. The propose hybrid system has been tested on a temperature control of a ceramic furnace plant.

Keywords: Knowledge engineering, PID, open-loop tuning, ruled-based system, hybrid intelligent system.

1 Introduction

This study tries to complete the research described in [1], where a Closed-Loop PID Controller design case was explained. Thus it becomes possible to analyze the PID empirical problem on its various possibilities. Also PID controller in its conventional form will be used [2] [3]. There is a large number of ways to obtain the parameters that define a controller [4-9], acquired by different means and under operating conditions that are peculiar to the equipment it is intended to control, as for example [29-34].

Despite previous research, practically in industry there is a large number of controllers in operation, whose tuning is far from what might be considered optimum [2]. This situation arises because, among other reasons, there is an important portion of users who lack familiarity with the adjustment techniques. It is necessary then to find new ways to solve this problem. Intelligent systems [10-13] can be a solution to it, due to the demand for enhanced performance and for the resolution of problems

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 352–362, 2011.

that are complex, either for humans or for machines. The time limits on decision-taking are increasingly tight and knowledge has become a major strategic resource in helping people to handle information and all the complexity that comes along. In an industrial context, hybrid intelligent systems are employed in the optimization of processes and systems relating to the control, diagnosis and rectification of problems [A14-A19].

Among the main advantages offered by knowledge-based systems are: permanence, duplication, speed, low cost, advantages in dangerous environments, reliability and explanation of reasoning. Development of knowledge-based systems is highly desirable in certain domains, and even essential in others.

There is the possibility of implementing simple knowledge-based systems by programming them into equipment. However, this does not take advantage of the specific tools available in the research area of Knowledge Engineering. In fact, knowledge-based systems available for handling control systems have some limited features [20-22].

One approach, to solve the above problems, is to create a generic decision method, based on a conceptual model describing the necessary steps to be achieved in order to obtain the PID controller parameters for open-loop empirical adjustment method. The novel model presented in this study was developed based on eight different sets of expressions with highly satisfactory results commonly used in control systems.

The rest of the paper is organized as follows: first, an explanation of the hybrid model proposed is given in section 2. This chapter is divided into two sub-sections: the first one (subsection 2.1) gives details about the deducted knowledge model based on experts contributions about controller tuning in open loop, the second one (subsection 2.2) shows how the new rules were obtained through a soft computing technique based on decision trees method [35] with the aim to complement the first one. Then section 3 presents an outline of the validation of the proposed technique. Finally the conclusions and future work are present.

2 Conceptual Modeling of a PID Controller

From the perspective of the human brain, a conceptual model of a domain consists of the most rigorous and explicit description of the knowledge of that domain. Thus, for the domain which is being dealt with in this study, a general summarized model is proposed in Figure 1.

As may be observed, this diagram is fundamentally divided into three blocks:

- Organizing Existing Rules. In this block, the intention is to organize the rules already available about types of expressions, ranges of application, and criteria for changes in load or monitoring the set-point, among others.
- Deducing New Rules to Complete the Knowledge Model. In this section it has become clear that there is a need to deduce new rules so as to build up a complete knowledge model from the system itself and the desired specifications until the parameters for the controller are finally obtained in a reasoned manner.
- Organizing Existing Knowledge with New Rules. This block is at the junction between the other two ones and is intended to order existing knowledge appropriately, for which it is necessary to draw up new rules.

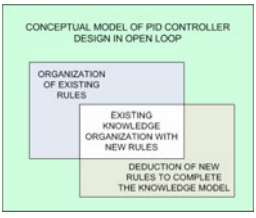


Fig. 1. General Summarized Overview of the Conceptual Model for Open-Loop

2.1 General Flow-Chart for Open-Loop Knowledge

In accordance with the guidelines established for the conceptual model, the conclusion is a general flow-chart of knowledge for the adjustment of the open-loop PID controllers, as shown in Figure 2, Figure 4 and Figure 5.

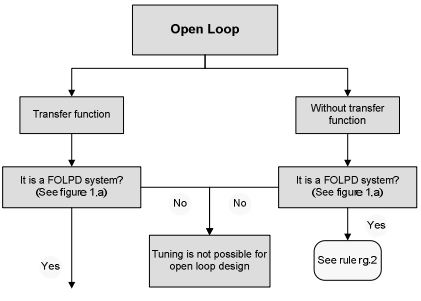


Fig. 2. First part of General Flow-Chart of Knowledge

In this part of the general flow-chart (figure 2), the first task to be done is to inquire whether the system is apt for PID open loop tuning methods. Then in both cases it is checked whether the response system is a first order lag plus time delay (FOLPD), if it is not the case, it would not be possible to carry out controller tuning with this method. In the positive case if the transfer function is not available it concludes with the rule "rg.2" (bottom-right of Figure 2). If the transfer function is known, the diagram will be followed.

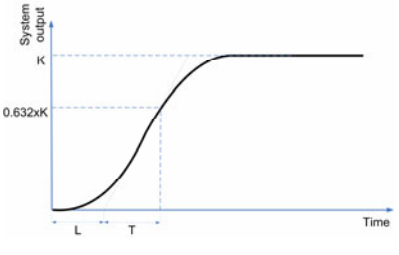


Fig. 3. System Response to step input

After having checked if it is a first order with time delay system response, and also the transfer function is known, it is necessary to follow the flow chart in Figure 4. First, the characteristics of the response "K" System Gain, "*L*" (Lag time) and "*T*" (response time) are obtained (figure 3) for a unit step input, and it is checked if the relation "*L/T*" is in the application range of the expression used in the study (between 0 and 1).

If it is not the case, this design method will not be applied. Otherwise L/T (dimensionless) is checked to see if it is bigger than 0.1 (empirical value [2]), if positive, it can be used to all the expressions contemplated in the study. If it is lower than 0.1 the question to ask the user is whether he/she wishes to discard the expressions of Ziegler-Nichols and Chien, Hrones, Reswick for not being within its scope of application range. If the expressions are not used, then rule "*rg.3*" (bottom-left of Figure 4) will be applied and, in any other case all methods will be taken into account as if L/T were bigger than 0.1. In any case, if all methods are contemplated, the next step is to determine a group with general characteristics to which the function of relation L/T belongs to, that will result with rule "rg.2".

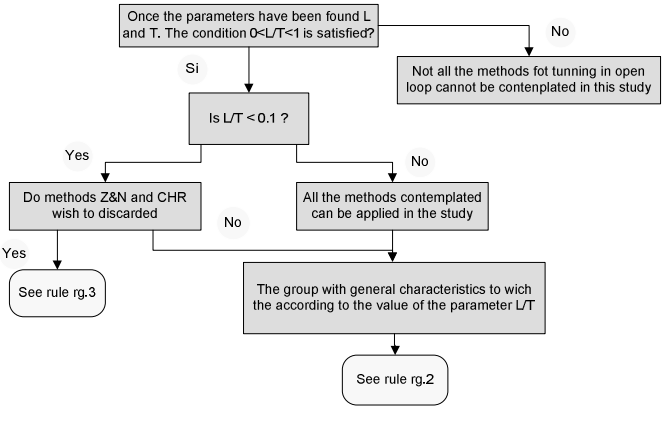


Fig. 4. Second part of General Flow-Chart of Knowledge

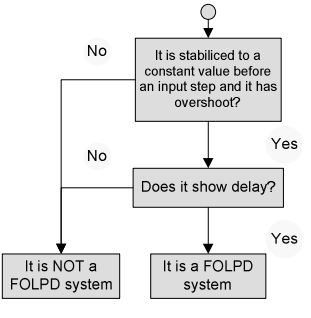


Fig. 5. General Flow-Chart to find out if a system is a FOLPD system type

The corresponding part of the diagram of figure 5 is employed to discover if the system response is a first order lag plus time delay (FOLPD) system. Two steps are necessary follow for it. The first step is to check if the system stabilizes at a constant value with a unit step input, and that there is no oscillation. If so, the next step is to make sure, if there is a system of FOLPD type. If both conditions are met, it can be concluded that the system is of this type, otherwise it is not.

2.2 Deducing Rules to Complete the Knowledge Model

As seen in the general summarized chart of knowledge (Fig.1), it is necessary to deduce new rules in order to complete the knowledge model. For this a need emerges to create a complete knowledge model, based on the system itself and the specifications desired, until the parameters for the controller are finally obtained in a reasoned approach. For this purpose, an example is given below to clarify how rules are deduced, using the ranges and new groups of representative systems [23] resulting of the processing and classification of the dataset obtained.

Deducing rule 2 (rg.2). This rule, as it can be observed in Figure 6, is applied in the most unfavorable case, in which the ratio L/T does not fall within the range of application of the equations. Figure 6 shows in the X axis representative systems included in [23] with a FOLPD response type, and in the Y axis the corresponding L/T parameter.

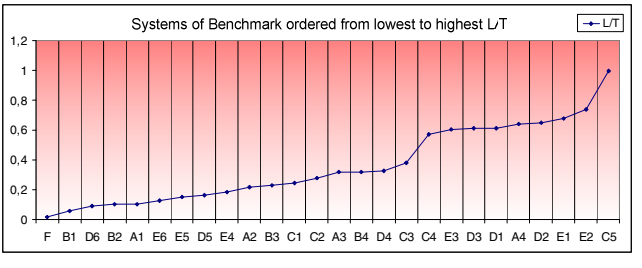


Fig. 6. Representative Systems (ax. X) ordered from Lowest to Highest Value for L/T (ax. Y)

Table 1 shows several representative systems [23] sorted by L/T value, where response is similar to figure 3. Table 2 shows the systems in the same order as in Table 1, and the best specification values (Tr, Ts, Mp and Tp) with the method used to obtain it such as Z & N (Ziegler and Nichols).

In order to create groups with generic characteristics, the various different systems are ordered from lower to higher value of L/T (Figure 5 and Table 1). Here, a table may be presented (Table 2), since the intention is to have generic groups for all specifications.

Table 1. Values for the L/T Parameter in representative Systems (F, B1, D6, ..) [23]

	System	L/T		System	L/T
1	F	0.0166	14	B4	0.318
2	B1	0.0582	15	D4	0.3269
3	D6	0.0982	16	C3	0.3797
4	B2	0.1021	17	C4	0.5729
5	A1	0.1036	18	E3	0.6054
6	E6	0.125	19	D3	0.613
7	E5	0.1493	20	D1	0.6141
8	D5	0.1649	21	A4	0.639
9	E4	0.1846	22	D2	0.6487
10	A2	0.217	23	E1	0.6763
11	B3	0.2299	24	E2	0.7377
12	C1	0.2464	25	C5	0.9964
13	C2	0.277			

The classification of these data will be achieved by means of decision trees [24-26], which are easily built into the expert system. The decision tree approach is one of the most common approaches in automatic learning and decision making. The true purpose of these decision trees is to classify the data in different groups, according to the dependent variable [24]. The decision trees have been obtained by using the J48 algorithm [24, 27, 28], using the software tool WEKA. The J48 algorithm was chosen because of its better performance in most circumstances [28]. Once the model was applied, a precision over the 80% was obtained for all the "*.arff*" (WEKA file extension) files created, corresponding to the different specifications covered, what means that the great amount of data have been correctly classified. In those cases where a correct classification was not achieved, a checking process is performed to see if the resulting data, obtained by the application of the method given by the tree, produces satisfactory results that do not differ much from the optimum method.

In Table 2, values for the specification are shown in each instance, along with the formulae for finding parameters used to improve this particular specification. Once the classification has been obtained, it is possible, for example, to group together the systems from D3 down to A4, within the conditio $0.6130 < L/T \leq 0.639$, establishing the following rules:

- In order to minimize Tr, apply the Ziegler and Nichols method.
- In order to minimize Ts, apply the CHR (Chien, Hrones, Reswick) 0% Mp (overshoot).
- To achieve the lowest percentage of Mp, apply the CHR 0% Mp.
- In order to minimize Tp, apply the CHR 20% Mp.

In spite of the systems D3 and D1, the best peak time result is obtained with ZN, if the rule for CHR is generalized to 20%, we achieve an error much smaller than if the system A4 is regulated with ZN. Nevertheless, this is the only true exception, so a generalization may thus be accepted; despite that it yields a small error.

3 Validation

The proposed conceptual model underwent a validation process. The validation process was not performed on those cases in which the transfer function was known and which perfectly fitted one of the systems related to the Benchmark [23]. Rather, it was carried out for the most unfavorable circumstances, which are produced in those cases where the transfer function is unknown or known but does not fit any of the systems.

Table 2. Rule 2 (rg.2) Groups for Changes in Load

System	Min. Resp. Time	Min. Stabl. Time	Min Overshoot	Min Peak Time
F	0,16 (Z&N)	1,64 (Z&N)	47% (CHR 0% Mp)	0,44 (Z&N)
B1	0,08 (Z&N)	1,01 (CHR 0% Mp)	45% (CHR 0% Mp)	0,23 (Z&N)
D6	0,53 (Z&N)	5,19 (CHR 20% Mp)	45% (CHR 0% Mp)	2,07 (Z&N)
B2	0,16 (Z&N)	2,08 (Z&N)	47% (CHR 0% Mp)	0,44 (Z&N)
A1	0,36 (Z&N)	5,29 (Z&N)	46% (CHR 0% Mp)	0,89 (Z&N)
E6	3,6 (Z&N)	54,01 (CHR 0% Mp)	46% (CHR 0% Mp)	9,96 (Z&N)
E5	1,84 (Z&N)	29,27 (CHR 0% Mp)	47% (CHR 0% Mp)	5,53 (Z&N)
D5	0,5 (Z&N)	6,57 (CHR 0% Mp)	42% (CHR 0% Mp)	2,01 (Z&N)
E4	0,77 (Z&N)	11,51 (Z&N)	46% (CHR 0% Mp)	2,52 (Z&N)
A2	0,81 (Z&N)	12,87 (CHR 0% Mp)	45% (CHR 0% Mp)	2,14 (Z&N)
B3	0,45 (Z&N)	5,88 (CHR 0% Mp)	45% (CHR 0% Mp)	1,26 (Z&N)
C1	0,8 (Z&N)	12,81 (CHR 0% Mp)	45% (CHR 0% Mp)	2,28 (Z&N)
C2	0,77 (Z&N)	13,08 (CHR 0% Mp)	43% (CHR 0% Mp)	2,39 (Z&N)
A3	1,22 (Z&N)	21,15 (CHR 0% Mp)	41% (CHR 0% Mp)	3,4 (Z&N)
B4	1,22 (Z&N)	21,15 (CHR 0% Mp)	40% (CHR 0% Mp)	3,4 (Z&N)
D4	0,43 (Z&N)	7,76 (CHR 0% Mp)	39% (CHR 0% Mp)	1,87 (Z&N)
C3	0,63 (Z&N)	13,13 (CHR 0% Mp)	39% (CHR 0% Mp)	2,59 (Z&N)
C4	0,39 (Z&N)	16 (CHR 0% Mp)	40% (CHR 0% Mp)	2,72 (Z&N)
E3	0,44 (Z&N)	14,05 (CHR 0% Mp)	32% (CHR 0% Mp)	1,87 (Z&N)
D3	0,27 (Z&N)	11,77 (CHR 0% Mp)	37% (CHR 0% Mp)	1,59 (Z&N)
D1	0,08 (Z&N)	8,19 (CHR 0% Mp)	46% (CHR 0% Mp)	1,09 (Z&N)
A4	2,67 (Z&N)	68,87 (CHR 0% Mp)	24% (CHR 0% Mp)	8,86 (CHR 20% Mp)
D2	0,15 (Z&N)	10 (CHR 0% Mp)	44% (CHR 0% Mp)	1,33 (Z&N)
E1	0,18 (CHR 20% Mp)	10,17 (CHR 0% Mp)	51% (CHR 0% Mp)	1,53 (CHR 0% Mp)
E2	0,41 (CHR 0% Mp)	13,69 (CHR 0% Mp)	37% (CHR 0% Mp)	1,75 (CHR 0% Mp)
C5	0,37 (CHR 0% Mp)	90 (CHR 0% Mp)	102% (CHR 0% Mp)	3,11 (CHR 0% Mp)

Table 3. Validation Results

Method indicated by the model	N. of cases	% of the total number of cases
The rule DOES coincide with the method that should in fact be used.	24 cases	66.6%
The rule DOES NOT coincide with the method that should in fact be used, but the deviation is very small.	7 cases	19.4%
It makes the system unstable.	4 cases	11.1%
The rule DOES NOT coincide with the method that should in fact be used, and the deviation is very considerable.	1 cases	2.7%

Validation was performed on nine systems not considered in the Benchmark. For each of them a check was made on all the specifications involved in the model. 36 cases were checked, for which the results shown in Table 3 were obtained. Thus, the novel model proposed is considered to have a satisfactory operation, since in overall terms their results are the following: Successes 86.1% and Failures 13.9%.

3.1 Empirical Verification on Ceramic Furnace

An empirical verification of the proposed novel method was performed on a small ceramic furnace (figure 7) in which the temperature is controlled. Essentially temperature depends on the power provided to heater element in the furnace. The controller is a modular Twido from Schneider with analog inputs and outputs modules. Temperature is measured in the furnace with a sensor based on thermocouple. The signal from thermocouple has been conditioned for a correct interpretation by programmable controller. For a good interaction of the user with the system the desired temperature is indicated with an external reference.

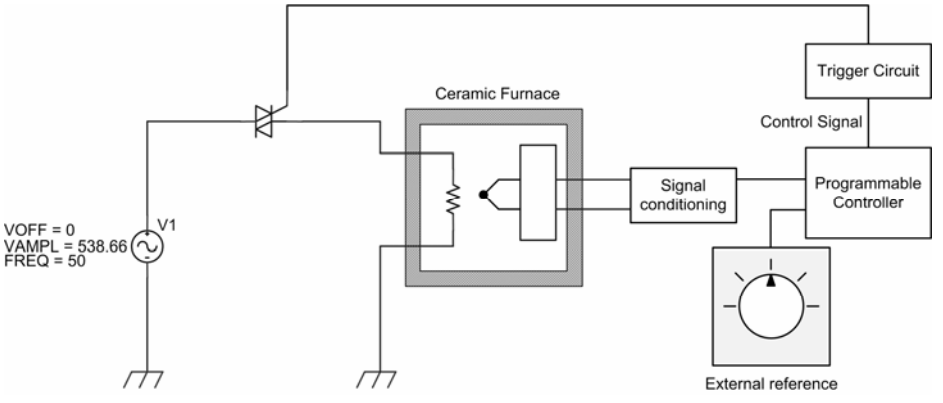


Fig. 7. Schematic diagram of the real plant

Initially the controller was programmed with Ziegler-Nichols tuning method in open loop. Then the novel hybrid generic decision method has been programmed on the controller. The program has been designed to run in two different methods: tuning mode and operation mode. Before starting the operation mode, the tuning step must take the parameters to the controller. In it, an external reference is fixed and the system waits until the temperature is stabilized according with the open loop method explained on section 2.1. For the present plant the system choose Ziegler-Nichols without overshoot method instead of Ziegler-Nichols traditional method. The experimental results are shown in figure 8.

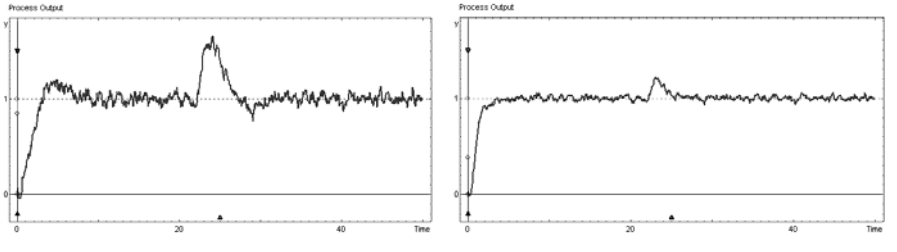


Fig. 8. Traditional Ziegler-Nichols method on the left and Ziegler-Nichols without Overshoot method on the right

4 Conclusions

The task of choosing the parameter adjustment method for PID controllers to be used may be obtained by using the technique proposed in this study. This is so even in the case where more than one method could be applied with satisfactory results.

After a choice is made about the method or methods to be used for finding the parameters, these are calculated in accordance with the procedure for the case that has previously been selected in a structured trend. In this way, the various possible routes that might be followed can be discriminated by rules, even to the extent of allowing a balance to be found between specifications that do not offer improvements of the same feature.

In producing a conceptual model, three useful contributions were made. Firstly, greater clarity was achieved for various stages in the adjustment of a PID design. Secondly, certain contradictions were shown up between different methods, these being resolved by the use of the model. Thirdly, a method for the automatic generation of new rules has been described, allowing to get a finer grain model based on data-mining and techniques.

The procedure was validated on real equipment whose transfer function is different from that linked to the reference used to obtain the model, for the most restrictive instances of the rules deduced. The results obtained and presented in the section relating to validation fulfill the initial objectives by verifying the correct behavior of the rules.

Acknowledgments. This research is partially supported by projects TIN2010-21272-C02-01 from the Spanish Ministry of Science and Innovation. The authors would also like to thank the manufacturer of components for vehicle interiors, Grupo Antolin Ingeniería, S.A. which provided support through MAGNO 2008 – 1028 – CENIT funded by the Spanish Ministry of Science and Innovation.

References

1. Calvo-Rolle, J.L., et al.: Development of a conceptual model for a knowledge-based system for the design of closed-loop PID controllers. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 58–65. Springer, Heidelberg (2009)
2. Astrom, K.J., Hagglund, T.: Advanced PID Control. ISA, Research Triangle Park. North Carolina (2006)
3. Feng, Y.L., Tan, K.C.: PIDeasyTM and automated generation of optimal PID controllers. In: Third Asia-Pacific Conference on Measurement and Control, China, pp. 29–33 (1998)
4. Chun-Fei, H., Guan-Ming, C., Tsu-Tian, L.: Robust intelligent tracking control with PID-type learning algorithm. Neurocomputing 71, 234–243 (2007)
5. Hung-Ching, L., Jui-Chi, C., Ming-Feng, Y.: Design and analysis of direct-action CMAC PID controller. Neurocomputing 70, 2615–2625 (2007)
6. Jun, Y.: Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot. Neurocomputing 71, 1561–1565 (2008)
7. Liu, H., Coghill, G.M.: A model-based approach to robot fault diagnosis. Knowledge-Based Systems 18, 225–233 (2005)

8. Sala, A., Cuenca, A., Salt, J.: A retunable PID multi-rate controller for a networked control system. Information Sciences 179, 2390–2402 (2009)
9. Gottwald, S.: Mathematical Fuzzy Control. A Survey of Some Recent Results. Logic Journal of IGPL 13, 525–541 (2005)
10. Yau-Tarng, J., Yun-Tien, C., Chih-Peng, H.: Design of fuzzy PID controllers using modified triangular membership functions. Information Sciences 178, 1325–1333 (2008)
11. Rodrigues-Sumar, R., Rodrigues-Coelho, A.A., dosSantos-Coelho, L.: Computational intelligence approach to PID controller design using the universal model. Information Sciences 180, 3980–3991 (2010)
12. Zhang, J., Zhuang, J., Du, H., Wang, S.: Self-organizing genetic algorithm based tuning of PID controllers. Information Sciences 179, 1007–1018 (2009)
13. Thangaraj, R., Chelliah, T., Pant, M., Abraham, A., Grosan, C.: Optimal gain tuning of PI speed controller in induction motor drives using particle swarm optimization. Logic Journal of IGPL (2010), doi:10.1093/jigpal/jzq031
14. Kareem-Jaradat, M.A., Langari, R.: A hybrid intelligent system for fault detection and sensor fusion. Applied Soft Computing 9, 415–422 (2009)
15. Karr, C.L.: Control of a phosphate processing plant via a synergistic architecture for adaptive, intelligent control. Engineering Applications of Artificial Intelligence 16, 21–30 (2003)
16. Hu, W., Starr, A.G., Zhou, Z., Leung, A.Y.T.: A systematic approach to integrated fault diagnosis of flexible manufacturing systems. International Journal of Machine Tools and Manufacture 40, 1587–1602 (2000)
17. Gholamian, M.R., Fatemi Ghomi, S.M.T., Ghazanfari, M.: A hybrid intelligent system for multiobjective decision making problems. Computers & Industrial Engineering 51, 26–43 (2006)
18. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72, 2729–2730 (2009)
19. Corchado, E., Abraham, A., Ferreira de Carvalho, A.C.: Hybrid intelligent algorithms and applications. Information Science 180, 2633–2634 (2010)
20. Wilson, D.I.: Towards intelligence in embedded PID controllers. In: Proceedings of the 8th IASTED Intelligent Systems & Control, Cambridge (2005)
21. Zhou, L., Li, X., Hu, T., Li, H.: Development of high-precision power supply based on expert self-tuning control. In: Control Systems and Robotics ICMIT 2005, Wuhan, China, pp. 60421T.1–60421T.6. (2005)
22. Epshtein, V.L.: Hypertext knowledge base for the control theory. Automation and Remote Control 61, 1928–1933 (2001)
23. Astrom, K.J., Hagglund, T.: Benchmark Systems for PID Control. In: Preprints FAC Workshop on Dig. Control. Past, Present and Future of PID Control, Tarrasa, pp. 181–182 (2000)
24. Parr, O.: Data Mining Cookbook. Modeling Data for Marketing, Risk, and Customer Relationship Management. John Wiley & Sons, Inc., New York (2001)
25. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, Inc., Chichester (2001)
26. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
27. Frank, E., Witten, I.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
28. Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 81–106 (1986)
29. Chun-Fei, H., Guan-Ming, C., Tsu-Tian, L.: Robust intelligent tracking control with PID-type learning algorithm. Neurocomputing 71, 234–243 (2007)

30. Hung-Ching, L., Jui-Chi, C., Ming-Feng, Y.: Design and analysis of direct-action CMAC PID controller. Neurocomputing 70, 2615–2625 (2007)
31. Rodrigues Sumar, R., Rodrigues Coelho, A., dos Santos Coelho, L.: Computational intelligence approach to PID controller design using the universal model. Information Sciences 180, 3980–3991 (2010)
32. Toscano, R.: Robust synthesis of a PID controller by uncertain multimodel approach. Information Sciences 177, 1441–1451 (2007)
33. Tsai, M.T., Tung, P.C., Chen, K.Y.: Experimental evaluations of proportional–integral–derivative type fuzzy controllers with parameter adaptive methods for an active magnetic bearing system. Expert Systems 28, 5–18 (2011)
34. Hwa Kim, D.: Hybrid GA-BF based intelligent PID controller tuning for AVR system. Applied Soft Computing 11, 11–22 (2011)
35. Verikas, A., Kalsyte, Z., Bacauskiene, M., Gelzinis, A.: Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: a survey. Soft Computing 14, 995–1010 (2010)

Clustering Ensemble for Spam Filtering

Santiago Porras[1], Bruno Baruque[1], Belén Vaquerizo[1], and Emilio Corchado[2]

[1] Civil Engineering Department,
University of Burgos
[2] Departamento de Informática y Automática,
Universidad de Salamanca

Abstract. One of the main problems that modern e-mail systems face is the management of the high degree of spam or junk mail they recieve. Those systems are expected to be able to distinguish between legitimate mail and spam; in order to present the final user as much interesting information as possible. This study presents a novel hybrid intelligent system using both unsupervised and supervised learning that can be easily adapted to be used in an individual or collaborative system. The system divides the spam filtering problem into two stages: firstly it divides the input data space into different similar parts. Then it generates several simple classifiers that are used to classify correctly messages that are contained in one of the parts previously determined. That way the efficiency of each classifier increases, as they can specialize in separate the spam from certain types of related messages. The hybrid system presented has been tested with a real e-mail data base and a comparison of its results with those obtained from other common classification methods is also included. This novel hybrid technique proves to be effective in the problem under study.

1 Introduction

One of the main problem that e-mail systems face nowadays is the management of spam. By this term we refer to the action of sending of indiscriminate unwanted e-mails, usually done in order to attract potential clients to use e-commerce systems to purchase articles or services or even to fraudulent services in order to obtain personal information that will be used afterwards for criminal activities.

To help users to cope with this flow of unwanted e-mails, almost every modern e-mail service includes a spam filtering service. This kind of filtering is ideally intended to automatically distinguish between spam and normal e-mails to present the final user only the wanted e-mail messages, freeing them of the task of classifying and eliminating those unwanted messages by themselves. The task of discerning which kind of messages are of interest to the final user is a complex task to perform in advance; so these systems must rely on a predefined configuration and try to adjust it to the preferences that each user provides to the system through his/her normal daily use. This kind of situation is clearly a task where automated learning can be of much use, as an automatic algorithm can

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 363–372, 2011.

be trained to perform this classification in a transparent way for the final user, by adapting its behaviour as it is used.

Many approaches exist to tackle this very common problem. According to what parts of the e-mail the filtering system analyzes, the solutions can be divided into header or meta-information based or content based. According to the architecture of the system a rough division could be between individual and collaborative systems. Usually, a modern spam filtering system would try to combine all of these techniques to benefit from the straights of each.

This study presents a novel hybrid intelligent system using both unsupervised and supervised learning that can be easily adapted to be used in an individual or collaborative system. It makes use of the Self-Organizing Map for a initial partitioning of data and the Naive Bayes for the final e-mail classification.

The rest of this paper is organized as follows: Section 2 presents a brief overview of the concept of ensemble learning, Section 3 includes a very simplified description of the SOM algorithm and more detailed explanations about its use a clustering algorithm. Section 4 introduces the hybrid model used for spam classification while Section 5 details the experiments performed with the algorithm and their results, compared with similar models. Finally conclusions and future work are described in Section 6.

2 Ensemble Learning

At its inception, the ensemble meta-algorithm [1] was created to improve the capabilities of existing models for data classification. The main concept behind ensemble learning model is the simple intuitive idea of a committee of experts working together to solve a problem.

The strength of the ensemble meta-algorithm is its potential to achieve a compromise between the desired result of both a small variance and a small bias; as a trade-off between fitting the data too closely (high variance) and not taking data into account at all (high bias). An important element is the effective combination of the classifiers, which relies in part on the presence of a certain variance in the components of the ensemble that is generally referred as 'diversity'. There is considerable evidence to suggest that the use of ensembles can lead to an improvement in the performance of single models in classification or regression tasks [2,3,4]. The underlying reason for increased reliability through the use of ensembles is that different classification algorithms will show different patterns of generalization. More formal explanations of the way ensembles can improve performance may be found in [5,6].

There are generally two ways to combine the decisions of classifiers in ensembles: classifier selection [7] and classifier fusion [6,8]. Classifier selection assumes that each classifier is an "expert" for some part of the feature space. In contrast, classifier fusion assumes that all classifiers are trained over the entire feature space. In the case of this work, the first approach is the one used.

3 Topology Preserving Maps

3.1 The SOM Algorithm

Topology preserving mapping comprises a family of techniques with a common target: to produce a low-dimensional representation of the training samples that preserves the topological properties of the input space. From among the various techniques, the best known is the Self-Organizing Map (SOM) algorithm [9]. SOM aims to provide a low-dimensional representation of multi-dimensional data sets while preserving the topological properties of the input space. The SOM algorithm is based on competitive unsupervised learning; an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, which are sets of samples in a specific domain of the input space. The update of neighbourhood neurons in SOM is expressed as:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v,k,t)(x(t) - w_k(t)) \tag{1}$$

where, x denotes the network input, w_k the characteristics vector of each neuron; α, is the learning rate of the algorithm; and $\eta(v,k,t)$ is the neighbourhood function, in which v represents the position of the winning neuron (Best Matching Unit or BMU) in the lattice, and k the positions of the neurons in its neighbourhood.

3.2 Data Clustering Capabilities

The clustering capabilities of the SOM algorithm have been extensively used in many works [10,11]. Mainly intended as a visual aid for data clusters exploration, several calculations over the final map have been proposed. After the calculation is performed, it can be represented over the final resulting map as a colour scheme. In the case of this study the three measures presented in [12,13] are used.

U-Matrix: This matrix represents for each unit of the map, the sum of the euclidean distances between its corresponding weights vector and all the data samples that are recognized by it. It permits to get a visual idea of how concentrated or disperse is the data set in the manifold represented by the map. Obviously, it is a valuable tool to detect and determine clusters and cluster borders in the data set.

P-Matrix: In this case, the values represented in the matrix are calculated as the density measured in the data space at a specific point, where that point is the weight vector associated with each unit of the SOM. As with previous matrix, the measure of the concentration of data serves as a very good measure for finding clusters in data.

U-Matrix:* This final matrix combines the distance based U-Matrix and the density based P-Matrix. It consists in using the U-Matrix as a basis for the final matrix, and the P-Matrix. Again, this matrix is used to obtain data clusters, especially in thin populated regions of the data space; where distances are more important that density to determine the similarity of data.

As the results obtained by all these three matrix calculations consist basically in numerical values for each of the units composing the maps, they can be used as an automatic way of determining the number of clusters in a data set but more importantly in this case, can inform about things like the boundary of those clusters or the density of data included in them. That information is used in the presented model to decide the structure that the ensemble of base classifiers will have.

4 Proposed Model

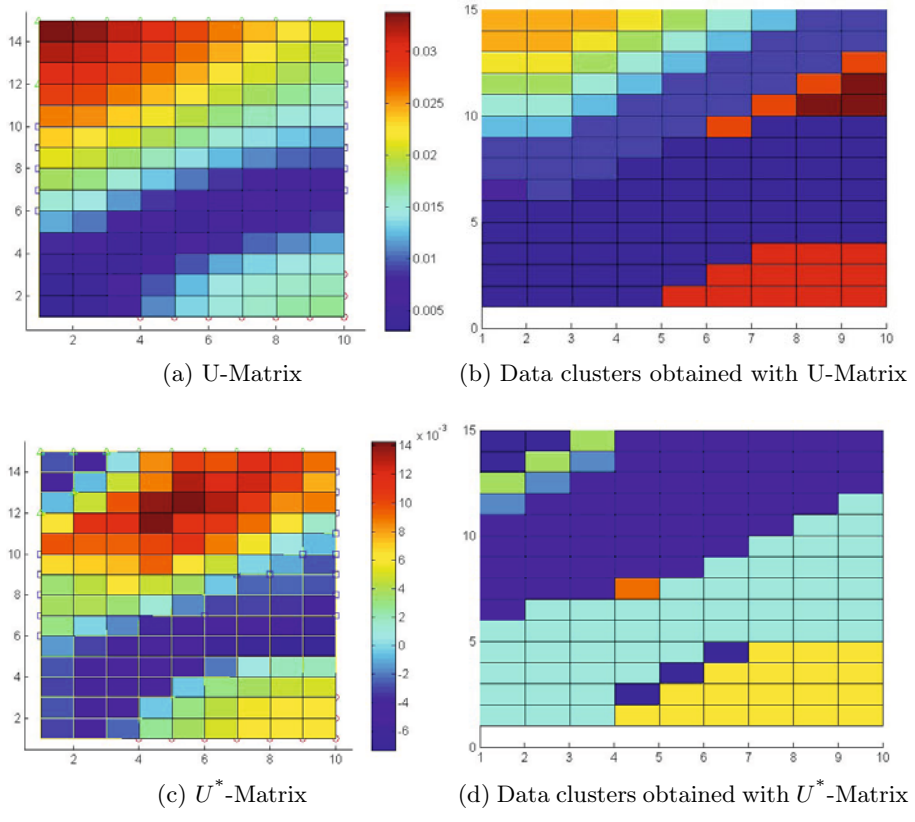
The intelligent hybrid model proposed in this research tries to take further the idea initially expressed in [14]. This previous work uses the well known k-nearest neighbours algorithm [15] as a way to initially explore the data set input space to latter construct an ensemble of classifiers that use that previously gather information to split data into the different components in a more informed way that other ensemble methods that use random or probability based distributions [16,17]. That way, an ensemble is constructed, ensuring that the base classifiers composing it will be expert in different regions of the data space.

In this case the SOM algorithm will be used to perform automatically this previous analysis and division of the data space. The use of this algorithm has several advantages over using other, simpler algorithms, such as the k-nearest neighbours.

In first place, the SOM is a complete grid extending over the data input space. That implies that is easy to observe and take into account not only similarity between samples, but to determine an ordered impression of that similarity, just using the distance between units on the map. Using the measures described in Section 3.2 several options for data clustering can be used, in a very straightforward way, to decide which configuration or architecture could be better for the construction of the ensemble and the data set partitions to train the model.

In the case of this study the SOM enables to split data between clusters, but also it is easy to find cluster frontiers were more detail and precision is required; therefore, data in those areas can be used to train several classifiers to enhance accuracy in those regions. For example, data situated in the borders of two clusters could be assigned to two different classifiers to try to obtain several different classifiers for those regions. Also, clusters that are either too extensive or that contain a high density of data can have more classifiers to back the classifications of others.

Other advantage that comes from the use of the SOM is that there is no need to use the number of clusters as a parameter to the algorithm, instead a threshold can be used to control how sensitive the algorithm is to changes in values for neighbouring units.



(a) U-Matrix (b) Data clusters obtained with U-Matrix

(c) U^*-Matrix (d) Data clusters obtained with U^*-Matrix

Fig. 1. Two of the mentioned matrix calculated on a 15x10 units SOM trained over the Iris data set

Figure 1 shows the values obtained for two of the matrix mentioned on Section 3.2 (Figs. 1a and 1c) and the corresponding clusters found with thresholds 0.0012 and 0.002 respectively (Figs. 1b and 1d) in the well-known case of the Iris data set [18]. Figs. 1a and 1c represent the values of the matrix in a color scale. It is easy to observe a gap with very low values that divides the data in two main groups. Figs. 1b and 1d depict the different clusters or divisions the algorithm has found in the data space. Each colour represents a different cluster.

The processing devised can be therefore summarized as in Algorithm 1.

5 Experiments and Results

The spam data used in this study is the SpamAssassin Public Corpus, published by the Apache Software Foundation [19]. Its main characteristics are: it includes instances of 6047 e-mail messages, with about a 31% spam ratio, sub-divided in three different classes “easy ham”, “hard ham” and “spam”. Much more detailed information can be found on the Internet public repository.

Algorithm 1. SOM Clustering and Selecting Ensemble

Input: A data set to be classified $D \epsilon \mathbb{R}^n$*, a clustering threshold* θ_c*, a inclusion threshold* θ_m

Output: A model able to classify novel entries $C_1 ... C_n$

procedure CONSTRUCT ENSEMBLE($C_1 ... C_n$)
    Train a Self-Organizing Map over the input data set.
    Label the units of the map with the entries for which it has been considered as the BMU.
    Calculate the requested matrix values for each map unit (U-Matrix, P-Matrix, U^*-Matrix).
    Perform a clustering of units depending in the difference in the value calculated in step 3 for each unit and its neighbours, using θ_c.
    Include each data sample in each of the clusters found in step 4, according to which cluster its corresponding BMU belongs to. Eliminate the clusters with a number of samples lower than θ_m.
    Train a base classifier with the data entries that form part of each of the clusters.
end procedure
procedure CLASSIFY SAMPLE(s)
    Present sample to the Self-Organizing Map and find the BMU
    Present sample to the classifier(s) corresponding to the BMU
    Output the majority vote of classifications obtained in the previous step
end procedure

5.1 Information Representation

In order to be able to work with the information contained in the e-mail is necessary to translate from plain text to a more manageable representation for an automated learning algorithm. This usually means to extract statistical characteristics form the analysis of the text, so each of the analyzed messages can be represented by an array of numerical values. This is a very widely known approach, basic in the discipline known as Information Retrieval (IR) [20,21]; used in tasks such as text clustering or classification, web searching, etc. In the case of this study, the codification used is the well-known "bag of words".

5.2 Experiments Description and Results

The experiments devised have the purpose of comparing different types of data partition methods to use for the ensemble construction. For all the five different partition schemes compared, the results are obtained from a 5-fold cross-validation process.

The five schemes are: the standard Bagging algorithm [16], which generates overlapping data sub-sets; and the k-means clustering algorithm and three variants of the presented model (U-Matrix, P-Matrix and U^*-Matrix), which generate disjoint sub-sets. The k-means clustering algorithm was previously used in [14] and is included here for the great parallelism with the presented model.

The tests have the same structure independently of the models chosen: first the data set is split between training and test sets. The training set is split into several sub-sets and a base classifier is trained over the data of each sub-set. All models used the classic Naive-Bayes classifier as their base classifier. Then, the test set is presented again to each model. The split model used to include training data in different sub-sets is used again to separate the test set and each sub-set is used as the inputs for their corresponding classifier.

Table 1. Parameters used for the SOM training in each case

Data set	Size (units)	Learning Rate	Epochs	Neighbourhood
Iris	8x8	0.1	1200	4
SPAM (25%)	15x15	0.1	10000	7
SPAM (complete)	30x30	0.1	15000	15

Table 2. Number of divisions used for each model and data set

			SOM		
Data set	Bagging	k-means	U-Matrix	P-Matrix	U^*-Matrix
Iris	5	5	13	4	1
SPAM (25%)	5	5	26	7	7
SPAM (complete)	5	5	58	27	15

Table 1 presents the parameters used for the SOM training in each of the data sets of the comparative.

It is interesting to note that the bagging and k-means methods require as inputs the number of folds/clusters in which the data set will be divided. This assumes that the user will have a relative knowledge about the data set. As explained before, the proposed algorithm does not need a so hard restriction over the algorithm, but only a difference threshold parameter; that lets the process to calculate the most adequate number of clusters. Table 2 shows the difference of the models in this regard.

Table 3. Percentage of classification error obtained using different techniques

			SOM		
Data set	Bagging	k-means	U-Matrix	P-Matrix	U^*-Matrix
Iris	4.66%	5.33%	**3.42%**	4.66%	4.66%
SPAM (25%)	19.1%	11.1%	10.17%	10.51%	**9.62%**
SPAM (complete)	50%	44.03%	39.54%	39.42%	**37.3%**

Table 3 shows the results obtained with the data set under analysis in this work. The Iris data set is one of the widely used for automatic learning test

and is included for comparative purposes. Finally, the two data sets object of the study are a fraction of the original Spam data set, used to try to avoid the computational complexity of training the model with the complete data set; and the complete one.

As results show, the use of hybrid system consisting on the SOM to cluster the data as a previous step to construct the ensemble can be generally regarded as an improvement over the other compared techniques. As hinted before, this situation comes from the fact that the SOM can provide a more detailed manifold to cluster the data, as opposed to the k-means, which is a much simpler technique.

When dealing with a very simple data set, as it is the case of the Iris data set, the extra complexity of calculating the SOM might not be so rewarding (the improvement is of less than a 2%). On the contrary, when working with bigger and more complex data sets (as the complete Spam, studied in this work) where there are large parts of the data set with overlapping classes, the more detailed divisions of the SOM can be of more clear use. In this study, the SOM obtains an error more than 5% lower than that of the k-means (see Table 3).

Regarding the different variants of the measures that can be calculated over the SOM, results seem to be better for the U-Matrix for the simpler data sets and for the U^*-Matrix for the more complex ones. This is result that will need further study.

6 Conclusions and Future Work

This study presents a model for the detection and filtering of spam based data on a hybrid intelligent system including both unsupervised and supervised learning. The model is variant of the idea of the clustering and selection concept, previously proposed by using a SOM as the clustering algorithm.

This novel hybrid model can be considered interesting for this task for several reasons. As explained in this work, one interesting topic on spam-filtering is the collaborative approaches, where several filtering systems work together to detect spam. The cluster and selection scheme studied here can be very easily adapted to work in these cases to distribute the classification among more specialized systems. This provides a heterogeneous framework where many models could be included. An additional reason is that the model can be used to provide the top-level mailing system administrator with complementary information, as the SOM included in the model has multi-dimensional data visualization as one of its main characteristics. Also, having an auto-adaptive model such as the SOM, which functioning can vary with the continuous training; could serve to soften the effect of concept drifts or modifications in the behaviour of spammers.

Regarding future work, it is interesting to mention that the meta-model presented here can be extended and improved in many ways, as it can be used as a framework for more complex schemes. As an example, in the test showed only a base classifier was trained for each of the clusters calculated. But due to the more detailed capabilities of the SOM for clustering, it could be adapted to train

classifiers for the data samples near the borders of clusters or to train several different classifiers to strengthen classification in clusters where classes overlap in a high degree.

Acknowledgments. This research has been partially supported through project of the Spanish Ministry of Science and Innovation TIN2010-21272-C02-01 (funded by the European Regional Development Fund). The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A., within the framework of the MAGNO2008 - 1028.- CENIT Project also funded by the MICINN.

References

1. Ruta, D., Gabrys, B.: An overview of classifier fusion methods. Computing and Information Systems 7(1), 1–10 (2000) (ISSN 1352-9404)
2. Schapire, R.E.: The strength of weak learnability. Machine Learning 5(2), 197–227 (1990)
3. Baruque, B., Corchado, E.: A weighted voting summarization of SOM ensembles. Data Mining andKnowledge Discovery 21, 398–426 (2010), doi:10.1007/s10618-009-0160-3
4. Corchado, E., Baruque, B.: Wevos-visom: An ensemble summarization algorithm for enhanced data visualization. Neurocomputing (in press, 2011)
5. Sharkey, A., Sharkey, N.: Combining diverse neural nets. Knowledge Engineering Review 12(3), 1–17 (1997)
6. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley Interscience, Hoboken (2004)
7. Jacobs, R., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation 3, 79–87 (1991)
8. Polikar, R.: Ensemble based systems in decision making. IEEE Circuits and Systems Magazine 6(3), 21–45 (2006)
9. Kohonen, T.: Self-Organizing Maps, vol. 30. Springer, Berlin (1995)
10. Lampinen, J., Oja, E.: Clustering properties of hierarchical self-organizing maps. Journal of Mathematical Imaging and Vision 2, 261–272 (1992)
11. Dara, R., Kremer, S.C., Stacey, D.A.: Clustering unlabelled data with SOMs improves classi cation of labelled real-world data. In: Proc. IEEE World Congress, on Computational Intelligence, pp. 2237–2242 (May 2002)
12. Ultsch, A.: Self-organizing neural networks for visualization and classification. In: Proc. Conf. Soc. for Information and Classification (1992)
13. Ultsch, A.: U*-matrix: A tool to visualize clusters in high dimensional data. Tech. rep., Department of Computer Science, University of Marburg (2003)
14. Kuncheva, L.I.: Clustering-and-selection model for classifier combination. In: KES, pp. 185–188 (2000)
15. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is nearest neighbor meaningful? In: Beeri, C., Bruneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 217–235. Springer, Heidelberg (1998)
16. Breiman, L.: Bagging predictors. In: Machine Learning, vol. 24(2), pp. 123–140 (1996)
17. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning, vol. 156, p. 148 (1996)

18. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
19. Apache Software Foundation. Spamassasin public corpus (2006)
20. Singhal, A.: Modern information retrieval: A brief overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24(4), 35–43 (2001)
21. Maron, M.E.: An historical note on the origins of probabilistic indexing. Information Processing and Management 44, 971–972 (2008)

Rule-Based Expert System Dedicated for Technological Applications

Edward Chlebus, Kamil Krot, and Michał Kuliberda

Institute of Production Engineering and Automation, Wrocław University of Technology
Łukasiewicza st 50-371 Wrocław, Poland

Abstract. This paper presents a description of an expert system developed specially for technological processes planning. Commercially available rule-based shell expert systems do not offer appropriate solutions in the scope of knowledge processing and knowledge representation. Hence authors had developed an own shell expert system devoid of aforementioned drawbacks. The most important property of the developed expert system is a hierarchical structure of technological knowledge base and a inference engine. By dint of implementation of above solutions it became possible aiding main stages of elaborating technological processes.

Keywords: expert system, CAPP, knowledge representation.

1 Introduction

Computer systems aiding process planning *(Computer Aided Process Planning)* are still in development. The most advanced CAPP systems are of generative type [1], [2]. In contrast to variant systems, where technological process planning for a new part takes place via modification of the same parts family representative's process, in generative system technological process is designed only on the basis of parts' specification and knowledge included in the system. Hence increasing emphasis is put on representation of technological knowledge included in CAPP system. This article presents results of work on knowledge-based, generative CAPP system designed in the Institute of Production Engineering and Automation of Wrocław University of Technology. Three fundamental modules of that system are presented:

- machining features recognition module,
- module of assigning permitted variants of technological process to recognised machining features,
- module of machining operations sequencing and setups planning.

For realisation of above modules a dedicated expert system had been designed. That system is compliant with rule-based paradigm. Moreover it has implemented knowledge processing mechanisms, which supports works related to technological process planning.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 373–380, 2011.

Structure of the knowledge base and special features of inference engine are described below. The article is concluded with the description of directions for further studies.

2 Expert Systems in Technological Processes Planning

The stages of technological processes planning which can be supported by use of expert systems are:

- selection of the form and material of a raw workpiece,
- recognition of machining features,
- determination of type of machining operations.

There are known studies where shell expert systems had been implemented into the area of technological processes planning [3]. However, such solutions are characterised by significant hindrances in representation of technological knowledge and its processing. It is a consequence of using an universal shell expert system, facilitating theoretically record of knowledge from distant fields of expertise. Shell expert systems are lacking functions allowing for:

- hierarchical representation of technological knowledge,
- management of facts' order in a knowledge base,
- formulation of complex mathematical functions within conditional parts of rules.

Due to the aforementioned limitations of rule-based shell expert system authors had developed their own system allowing for complete representation and processing of technological knowledge.

3 Input Data for a CAPP System

In developed system parameters of a new part are imported from the CAD 3D SolidWorks system. That system is using a boundary representation (*B-rep)* for geometrical models. Furthermore it allows for model completion with technological characteristics such as: dimensional and geometric tolerances, surface roughness, part's material, datums etc. By dint of these properties part's models created in SolidWorks can be used as input data for a CAPP system.

A model in *B-rep* consists of faces bounded by edges, which in turn are described by vertices. Fundamental model elements in boundary representations are presented in fig. 1.

A face is a fragment of a surface bounded by loop. A loop in turn is a set of curves. There are inner loops i.e. those which are excluding a certain area within a wall and outer loops which are limiting the face from the outside. Each loop consists of curves, which in the SolidWorks system can be segments, arcs or Bezier curves. Curves are parameterised by means of determining their type and set of vertices defining them.

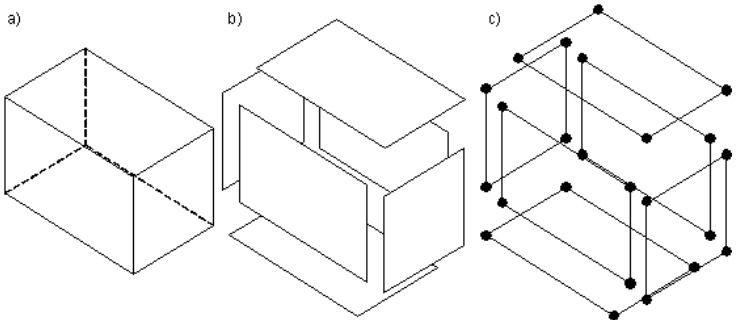


Fig. 1. 3D model elements in boundary representation: model overview (a), model faces (b), faces' edges and vertices (c) [4]

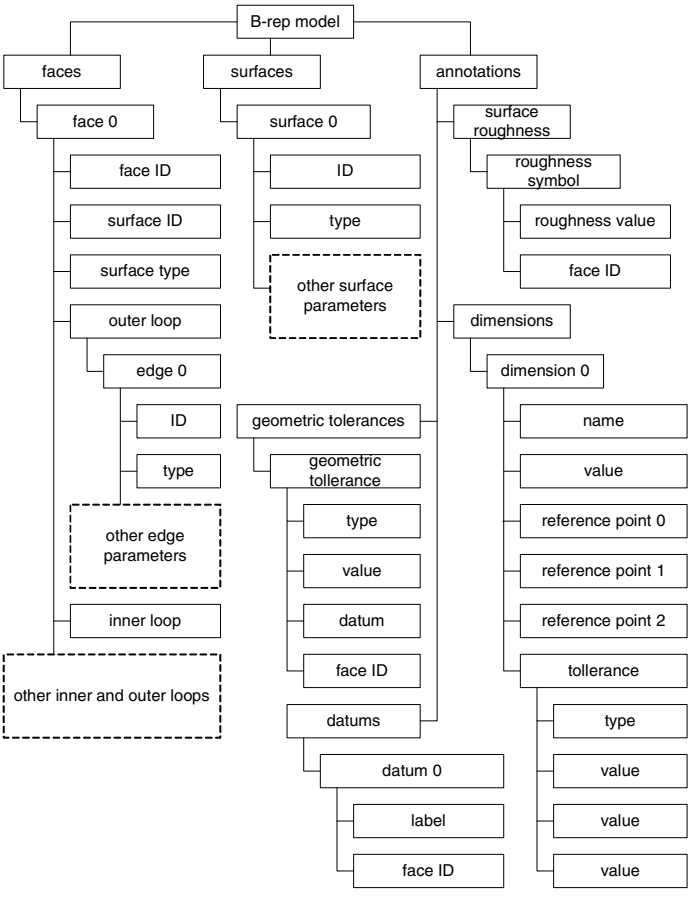


Fig. 2. Hierarchical structure of a knowledge base representing a CAD 3D model's parameters[5]

4 The Knowledge Base Structure

A knowledge base of a rule-based expert systems consists of facts and rules. Facts consist of trios <O, A, V> - object attribute and values or duos <A, V> - attribute and value only. In such representation an object denotes usually actual subject of analysis or diagnosis, an attribute is its feature, and a value is the measure of that feature expressed as a number or symbolically depending on the attribute's type. That representation is sufficient for typical diagnostic cases or for solving easy due to dependencies between parameters decisional problems. Unfortunately in case of processes planning such knowledge representation is insufficient. It is necessary to process many data of the same kind, e.g. hundreds of a model's faces described by means of many parameters, moreover holding geometrical objects of another type. Part's model and the structure of technological process characterise by hierarchical dependencies. Therefore the knowledge base of the developed expert system allows building dependencies of such type. An exemplary structure representing a part's model has been presented in fig. 2.

Hierarchical structure of facts allows for representation, complex in their description CAD models. It allows also for compliant with reality description of dependencies of the parent-child type.

5 Rule-Based Machining Features Recognition

Under the term of feature is understood the general shape of characteristics of a product, with which certain traits and knowledge can be linked [4]. In the set of features we can distinguish, most frequently used in CAD systems, machining features, defined as a shape traced by the cutting tool's edge in the machining process. Hence machining objects are always subtracted (they mean removal of a part's fragment), in contrast to e.g. design objects, which can be both subtracted and added. Obtainment of machining features from CAD models is the fundamental task regarding integration of CAD and CAPP systems. Those objects can be obtained by recognition in existent CAD models [6], [7], [8], [9], [10] or by designing a CAD model using machining features. The second technique did not catch on in practice due to restrictions it imposes upon model designer, thus recognition is currently the most commonly used technique of obtaining machining features. This technique can be further split into automatic and interactive. Automatic recognition assumes lack of user input into the process, contrary to interactive recognition, where the user indicates appropriate elements of a part's model and creates a feature out of them, and the system verifies the correctness of the user's choice. In practice though, only the automatic recognition complies with assumptions of generative CAPP systems and only that technique is currently developed. In the developed system a rule-based approach to features recognition has been implemented. However the structure of rules and supporting them inference engine are significantly different from typical decision rules and forward/ backward chaining. The conditional part of the rule of such type includes a list of faces' parameters and their mutual relations which ought to be satisfied, to identify the machining feature of a predetermined type on their basis.

The working principle of this rule is presented on the example of recognition of a typical machining feature being a pocket from fig. 3.

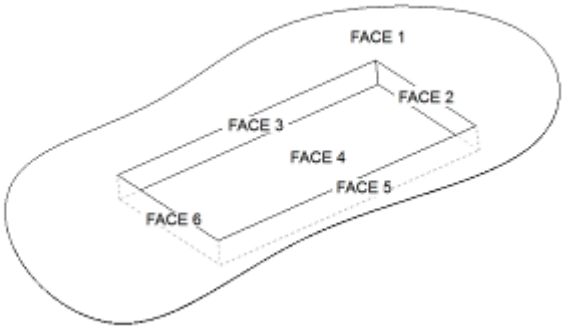


Fig. 3. Faces building a machining feature of pocket class

Below is presented an exemplary set of recognition conditions for the pocket from fig. 3:

- parameters of faces:
  - face 1 – planar, can include inner loops, does not constitute an object,
  - face 2, 3, 5, 6 –planar, cannot include inner loops, constitutes an object,
  - face 4 – planar, can have inner loops, constitutes an object,
- relations between faces:
  - face 2 – face 3 – perpendicularity, normals facing centre,
  - face 2 – face 5 – perpendicularity, normals facing centre,
  - face 5 – face 6 – perpendicularity, normals facing centre,
  - face 5 – face 3 – perpendicularity, normals facing centre,
  - face 2 – face 4 – shared edge,
  - face 3 – face 4 – shared edge,
  - face 5 – face 4 – shared edge,
  - face 6 – face 4 – shared edge,
  - face 2 – face 6 – shared edge,
  - face 3 – face 6 – shared edge,
  - face 5 – face 6 – shared edge,
  - face 6 – face 6 – shared edge,
  - face 3 – face 4 – perpendicularity,
  - face 1 – face 3 – perpendicularity.

The effect of executing this rule is adding an appropriate number of machining features to the fact's knowledge base.

6 Assigning Variants of Machining Process into Machining Features

Under similar rule to the features' recognition mechanism works the mechanism of assigning variants of machining process to previously recognised machining features. Usually a single machining feature can have assigned to it several possible variant of a technological process. The identification of machining operations is a process of adding fragments of a hierarchical structure with a single root object to an appropriate location of the knowledge base's structure. Machining operations and their parameters are added as child objects to objects representing machining features. The input data to the machining operations determination process, are parameters of a machining feature. The pseudocode of an exemplary rule of that type is presented in listing 1.

Listing 1. A rule identifying machining operation of through holes

```
If there is
object through hole* being descendent of the MF
(machining feature) object
of attributes satisfying the following dependencies:
diameter ≥ 1.5
depth / length ≤ 10
accuracy class == 11
then
add to the through hole*object
the process variant object, including:
the Operation 1 object with attribute of Type ==
drilling and Diameter == 1.4
```

7 Sequencing of Machining Features

It had been adopted in assumptions of the developed system, that the order of machining features in the knowledge base is reflecting the machining. Originally machining features are arranged in the way that the earliest added objects are occupying the very beginning of the list, while at the end are objects added last. Such order normally does not correspond to order of machining. At this point the distinction between critical sequence and optimum sequence needs to be made. The first of those means the machining features' sequence, which enables realisation of a technological process ensuring obtainment of assumed dimension and shape precision. In turn the optimum sequence takes into account previous constraints and additionally allows for process realisation in accordance with assumed optimisation criteria, e.g. minimisation of production cost and time. Determination of critical machining sequence only, can be achieved by use of rules. The optimum sequence requires application of different techniques such as e.g. genetic algorithms and is going to be the subject matter of further research.

8 Graphical User Interface of the Developed System

Figure 4 presents the main window of the developed system.

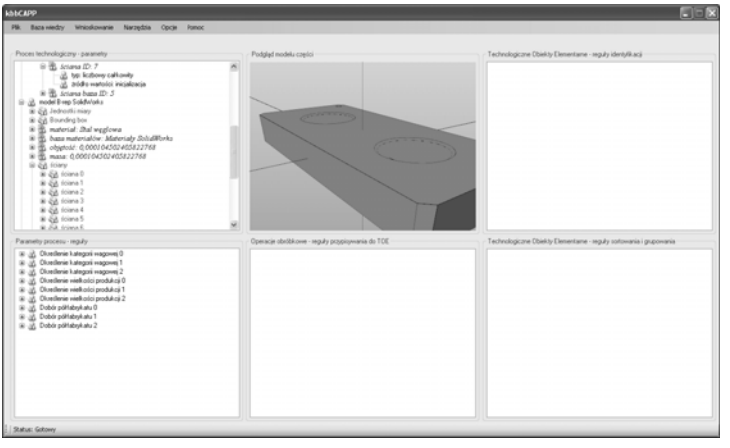


Fig. 4. The main window of the developed CAPP system

This window is divided into six panels. The panel in the top left shows the fact's knowledge base structure. The panel in the top middle in turn presents 3D view of a part's model. The four remaining panels display rules of following types:

- in the top right panel – rules responsible for recognition machining features,
- in the down left panel – common decision rules known from rule-based expert systems, responsible in this case for raw workpiece selection,
- in the down middle panel – rules responsible for selection of machining operations,
- and the down right panel – rules responsible for sequencing machining features and setups planning.

New facts and rules can be added using context menu. Moreover this menu allows changing parameters of existing fasts and rules.

9 Summary

The article presents the current state of development of a knowledge based, generative CAPP system. Modules realising following functions has been developed:

- machining features recognition,
- assigning machining operations to previously recognised features,
- determining the critical machining features' sequence,
- setups planning.

Current authors' research interests concentrate on the development of a machining operations sequence optimisation module.

References

1. Zhou, X., Qiu, Y., Hua, G., Wang, H., Ruan, X.: A feasible approach to the integration of CAD and CAPP. Computer-Aided Design 39, 324–338 (2007)
2. Park, S.C.: Knowledge capturing methodology in process planning. Computer-Aided Design 35, 1109–1117 (2003)
3. Wang, K.: An integrated intelligent process planning system (IIPPS) for machining. Journal of Intelligent Manufacturing 9, 503–514 (1998)
4. Shah, J., Mäntylä, M.: Parametric and feature-based CAD/CAM. John Wiley & Sons, Chichester (1995)
5. Kuliberda, M.: Elaboration of selected modules of the knowledge-based generative CAPP system. PhD Thesis : Institute of Production Engineering and Automation of Wrocław University of Technology (2009)
6. Brousseau, E., Dimov, S., Setchi, R.: Knowledge acquisition techniques for feature recognition in CAD models. Journal of Intelligent Manufacturing 19, 21–32 (2008)
7. Ismail, N., Abu Bakar, N., Juri, A.H.: Feature Recognition Patterns for Form Features Using Boundary Repr. Models. The International Journal of Advanced Manufacturing Technology 20, 553–556 (2002)
8. Marchetta, M.G., Forradellas, R.Q.: An artificial intelligence planning approach to manufacturing feature recognition. Computer-Aided Design, 248–256 (2010)
9. Babic, B., Nesic, N., Miljkovic, Z.: A review of automated feature recognition with rule-based pattern recognition. Computers in Industry 59 (2008)
10. Krot, K., Kuliberda, M.: Technological features identification of machine parts in CAD 3D geometrical models using boundary representation. In: Production Engineering Knowledge-Vision-Framework Programmes, Wroclaw University of Technology (2006)

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

Edward Chlebus, Anna Burduk, and Arkadiusz Kowalski

Wrocław University of Technology, 27 Wybrzeże Wyspiańskiego St,
50-370 Wrocław, Poland

Abstract. The simulative agent system is going to be capable of independent collection of information required for construction of a simulation model, monitoring and constant data updating, deciding on the necessity of building new updated simulation models, applying changes to data (based on results of a conducted simulation) into class ERP and PDM systems. In this way will be achieved – apart from faster model construction – an increase in reality reproduction precision, in comparison to simulation models built with traditional methods.

Keywords: agent system, manufacturing system model, data exchange.

1 Introduction

The premise prompting to take on the subject of ways of building simulation models is the fact, that the majority of time and effort is consumed by data gathering, their analysis and by model constructing – Fig. 1. In case when insufficient time is being devoted to adequate model planning, a simulation project becomes difficult and complicated.

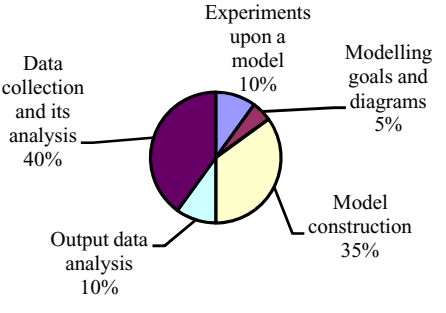


Fig. 1. Percentage distribution of time intended for individual phases of a simulation project [8], updated in 2009

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 381–388, 2011.

The success of a simulation project is dependent on simulation and project management tools, but also on acquisition of appropriate information, scattered usually across different enterprise departments. Construction of a model – given pertinent tools are used – takes up only approx. 35 percent of the time allocated to an optimisation project of a manufacturing system.

1.1 Ways of Obtaining Data for Building Simulation Models

Remedy for this situation is application in data gathering and analysing data exchange agents, which by means of appropriate interfaces would explore databases in ERP, SFC, PDM systems, operating within an enterprise, in order to discover and garner a diverse range of data requisite for building a simulation model of a manufacturing system. In that manner the phase of data gathering and analysis is omitted – approx. 40% saving of time devoted currently to project realisation.

The next crucial step would be to create a possibility of automating construction of a simulation model and subjecting it to simulation. Model building phase would have been also substantially shortened – it would be limited only to end user accepting gathered data and the obtained model.

The user would obtain a simulation model of a manufacturing system for further analyses and optimisation activities, as well as a possibility of implementing gained results back to an ERP system – in that manner functioning of the CRP module would have been improved – which so far boils down to manual estimation of planned orders' influence on available production capacities in a middle-tem planning horizon. Hence it is possible to exceed workcells' performance on a daily basis. This module uses for its analyses data from a MRP report. Analysis of routings and current situation in production departments allows for calculating production capacities to end work of each particular work centre.

Possible improvements in ERP systems' functioning in the area of production are essential – seeing how little implementations of ERP systems are successful, by which is understood activation of the MRP loop and modules aiding strictly production activities.

In case of simulation models being built for further analyses and optimisation, amongst user tasks are enumerated only determination of time and scope of its construction, whereas data exchange agent will provide a set of representative data for a model of a department, workshop or enterprises' holding connected with a common ERP system. In that way, an increase in reproduction's accuracy can be achieved, in comparison with simulation models built in traditional manners.

1.2 Main Tasks of a Software Agent

Main tasks of a software agent will include:

- gathering information on components of a production system,
- presentation of acquired data in a form possible to import into a simulation system, featuring various accuracy levels of the proposed model,
- independent monitoring of occurring changes in gathered information, making decisions on updating a simulation model of production processes in an manufacturing company.

The simulative agent system is going to be capable of:

- independent collection of information required for construction of a simulation model,
- monitoring and constant data updating, deciding on the necessity of building new updated simulation models,
- applying changes to data (based on results of a conducted simulation) into class ERP and PDM systems.
- influencing functioning of chosen ERP system modules e.g. the CRP module.

The next premise in favour of project realisation is growing degree of complexity of today's manufacturing systems. It confirms the necessity of building interfaces between ERP systems and simulation tools.

A system designed for simulation of manufacturing systems will support a production engineer in terms of simulative analysis in the following areas:

- product development and project management – currently simulative analyses and tools are the most important methodology in implementation of a new product into production,
- simulation and visualisation of the course of a production order,
- material flows, transport routing planning,
- plans of arranging work stations (Layout),
- estimation of a manufacturing process's cost-intensiveness.

Proposed diagram of simulation engine communication with environment in a system intended for simulation of discrete processes is depicted in Fig. 2

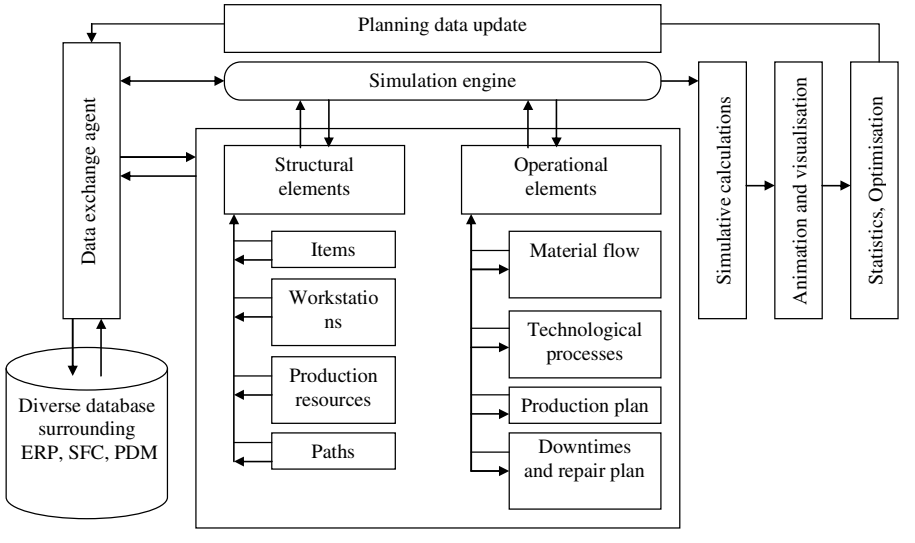


Fig. 2. Diagram of Communication in a system with agents

In the field of developing agent programming theory answers are sought to fundamental questions:

- How to build agents capable of independent, autonomous functioning, so that they would auspiciously execute delegated to them tasks?
- How to build agents capable of interaction (cooperation, rule, negotiating) with other agents in order to auspiciously execute delegated to them tasks?

2 Agent Programming Theory Development

The subject of software agents turns out to be a young field of knowledge, indicative of which is at least the fact, that thus far there has been no approved definition of software agent elaborated, hence further scientific research into the notions' unification should be expected, as well as determination of features characterising an agent. There is no commonly accepted agent typology either.

There are various definitions of a software agent, however so far no uniform and coherent its concept has been developed. Presented below are several approaches:

- An agent is anything, what can be acknowledged as observing surrounding by sensors and functioning within that surrounding via effectors [11],
- Closed computer system located within a certain surrounding, having an ability of flexible functioning in that surrounding, consisting of fulfilling objectives intended for its creation [13].

An agent is perceived then, as a computer system capable of making self-reliant, independent actions in order to achieve a predetermined aim. By multi-agent system we understand a system composed of many mutually cooperating agents. Fundamental features of every agent should be:

- autonomy of decisions made and actions taken,
- ability of cooperating with other agents or coordination of their tasks, capability of negotiating with other agents,
- capability of observing changes taking place in agent's environment and responding to them,
- activity in achieving goals, exploiting favourable conditions and initiative taking,
- capability of learning, adopting to new situations based on hitherto inferences [5], [13].

Software agent as a method of creating software has been evolving from programming in machine code to object-oriented programming - the set of features characterising software agents has not been fully established.

By analysing authoritative materials of the annual WSC conference (Winter Simulation Conference, USA), the general interest increases in utilisation of software agents and multi-agent systems (MAS) in the field of computer simulation. Year on year the number of publications increases, what is especially visible in fields related to the theory of agent simulation models' functioning and military themes:

- software development – next generation simulation: [4], [9], [10],
- subjects related to simulation utilisation in the area of military: [7], [3], [2],
- agent modelling: [6], [12], [7].

Advantages of software agents and multi-agent systems:

- agents constitute the best answer to rapid growth of available information, also one scattered across internet,
- agent systems constitute the best platform possible for building intelligent adaptive systems,
- mobile agent systems are the pre-eminent platform supporting mobile users – mobile agents can move between computers what facilitates much better utilisation of available resources,
- agent systems have got the potential to become the new methodology of creating software,
- possibility of creating copies of agents, what facilitates introduction of redundancy into the system, and thus limitation of its failure rate.

As with every other new approach, the theory of agent systems has got its drawbacks:

- whatever can be developed and implemented in form of an agent system, can be programmed in another way either,
- thus far there are no prominent implementations of agent systems,
- the theory of agent systems has been dysfunctional hitherto in developing a precise definition of a software agent and agent systems,
- agent is a potential mobile Trojan horse able to visit many computers.

For a project realisation, available systems for creating agent software can be used. There is a vast choice amongst that type of software, unfortunately though, only a part of them is continued to be developed – Table 1, at [1].

Table 1. Overview of chosen systems for creating agent software

1. Agentbuilder	11. Gossip 1.01	21. Kaariboga
2. Aglets	12. Grasshopper	22. MadKit
3. Ajanta	13. Gypsy	23. Mast
4. Bee-gent	14. Hive	24. Mole
5. Bond	15. IBM-Able	25. OAA 2.0.9
6. Concordia 4)	16. JACK	26. Pathwalker 1.0
7. Cougaar	17. Jade	27. Ronin 1.1
8. D'Agents 2.0	18. Jafmas	28. Soma 2.0/3.0 Beta
9. FarGo	19. Jess	29. Voyager 3.3
10. FIPA-OS	20. Jumping Beans	30. Zeus 1.02

Creation of agent systems is connected with creating software for highly convoluted systems and requires provision of cooperation between technologies, which were being created independently of each other, hence at the moment are unprepared for cooperation. It is indeed the level of complexity of the task facing creators of software based on agent systems, which is one of major causes for low number of implemented test systems up to this point.

From the point of view of enterprise management needs, the greatest usefulness is characterising currently simulation systems, based on the concept of modular graphical implementations. One should assume, that construction of a simulation model will not demand from the end user knowledge of a programming language, what should increase the number of interested enterprises.

3 Classification and Description of Objects in a Simulation Model of Manufacturing Processes

The proposed simulation system will include the following classes of objects, corresponding to structural elements of a manufacturing system model (Fig. 3): MP items – objects manufactured products are composed of in manufacturing processes. The most common use of MP is expected to represent in a group's model elements being in a container during transportation activities.

- MT machines and devices – stations where objects are processed, assembled and stored etc. Depending on the level of detail of model, MT can present a production line, assembly cell, machining centre, single workstation or assembly table.
- MZP resources – personnel, means of transport used in the manufacturing process. To MZP can be assigned resource transportation time and by its means presented consumed resources, e.g. material (welding wire) and immaterial (compressed air).
- MS paths – objects' or resources' transportation routes within a system. Thanks to differentiation of MS various kinds of transport routes can be taken into account, depending on resources moving along them.

Main tasks requisite for realisation of a proposed agent system include:

- comparative analysis (benchmarking) of usefulness for application of different algorithms and discrete simulation functioning methods,
- development of parameterised, graphical object libraries of technical infrastructure objects,
- development of algorithms of data exchange agents' operating, describing rules of cooperation and possibility of building simulation model's objects of different detail levels,
- development of database of manufacturing system model's constituents, including processed items, workstations, production resources, paths,
- development of data exchange agents enabling data import,
- development of procedures and algorithms for construction rules of simulation models,
- development of algorithms and procedures for discrete simulation engine,
- development of algorithms and modules of a system enabling animation of a manufacturing system's simulation model,
- development of a module for analysis of results obtained from a conducted simulation of a manufacturing system model,
- integration of developed modules of a system and their testing,
- elaboration of obtained results of project's final report.

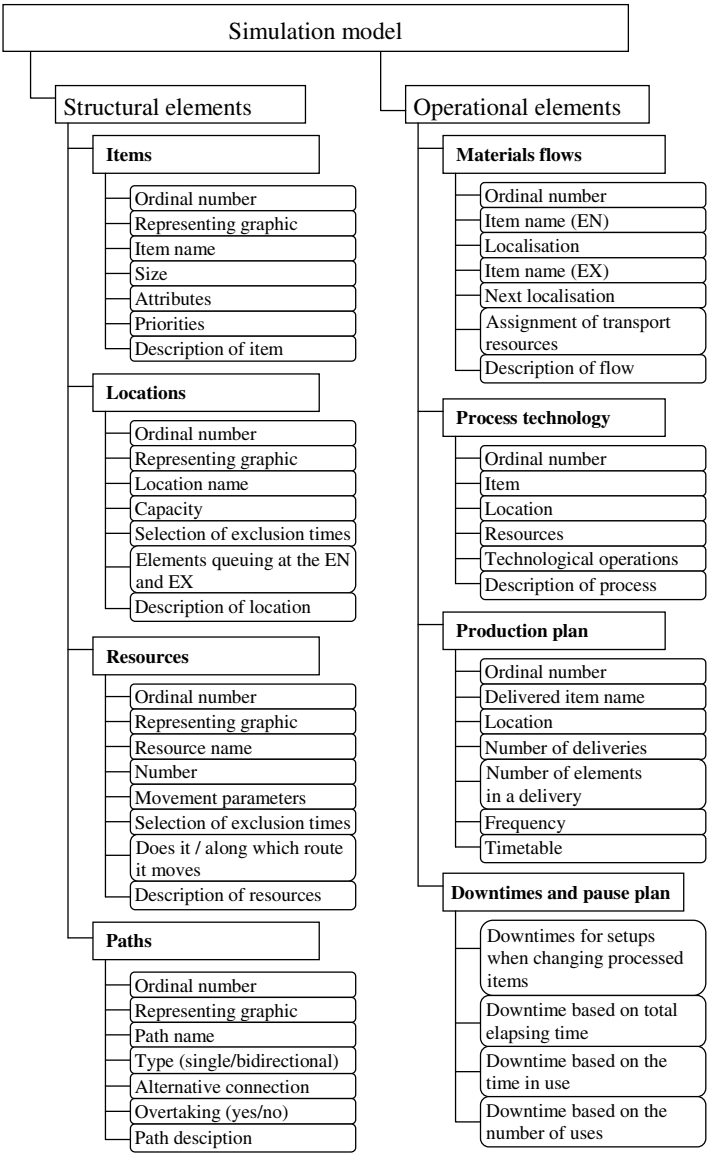


Fig. 3. Overview of data about structural and operational elements constituting a simulation model for mass production

4 Summary

In order to remove diagnosed in this article drawbacks of using simulation tools, data exchange agent implementation is suggested, which would provide collection of data

representative of a model of department, facility or holding of enterprises connected with a joint ERP system. In this way will be achieved – apart from faster model construction – an increase in reality reproduction precision, in comparison to simulation models built with traditional methods.

The main tasks of a software agent will include: gathering information on production system components, presentation of collected information in from possible to be imported into a simulation system, taking into account different levels of detail of proposed model, self-reliant monitoring of changes taking place in the accumulated information, decision making about updating a simulation model of production processes in a manufacturing company.

References

1. Altmann, J., Gruber, F., Klug, L., Stockner, W., Weippl, E.: Using Mobile Agents in real World: A Survey and Evaluation of Agent Platforms. In: Proceedings of the Second International Workshop on Infrastructure for MAS, and Scalable MAS, Montreal (2001)
2. Champagne, L.: Bay of Biscay: Extensions into Modern Military Issues. WSC (2003)
3. Ganapathy, S., Hill, R.: Dynamic Path-Planning for Search and Destroy Missions – The Bay of Biscay Scenario. WSC (2003)
4. Himmelspach, J., Röhl, M., Uhrmacher, A.: Simulation for Testing Software Agents – An Exploration based on JAMES. WSC (2003)
5. Jennings, N., Wooldridge, M. (eds.): Agent Technology: foundations, applications, and markets. Springer, Heidelberg (2002)
6. King, G., Heeringa, B., Westbrook, D., Catalano, J., Cohen, P.: Models of Defeat. WCS (2002)
7. Mizuta, H., Yamagata, Y.: Agent-Based Simulation and Greenhouse Gas Emissions Trading. WSC (2001)
8. Nordgren, B.W.: Steps for proper simulation project management. In: Winter Simulation Conference, Association for Computing Machinery (1995)
9. Pathak, S., Dilts, D., Biswas, G.: A Multi-Paradigm Simulator for Simulating Complex Adaptive Supply Chain Networks. WSC (2003)
10. Riley, P., Riley, G.: SPADES – A Distributed Agent Simulation Environment with Software-in-the-Loop Execution. WSC (2003)
11. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall, Inc., Englewood Cliffs (1995)
12. Whelan, M., Loftus, J., Perme, D., Baldwin, R.: Reducing Training Costs through Integration of Simulations, C4I Systems, and Expert Systems. WCS (2002)
13. Wooldridge, M.: Intelligent Agents. Springer, Heidelberg (1997)
14. Wooldridge, M.: An Introduction to MutiAgent Systems. John Wiley & Sons, Chichester (2002)

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

Anna Burduk

Wrocław University of Technology, ul. Wybrzeze St Wyspianskiego 27,
50-370 Wrocław, Poland
anna.burduk@pwr.wroc.pl

Abstract. Proposed in literature quantitative methods of risk analysis and evaluation treat single issues, assuming certain factors and conditions as well as impose constraints. Hence in order to asses risk of production process in its real environment, the problem should be simplified and adjust to a certain method. Taking into consideration the complexity of modern production systems as well as a number of influencing them external, random factors, this kind of approach seems to be unsuitable. This paper presents a method of determining the risk for a production system and a coefficient of changes' risk.

Keywords: Risk, production system, reliability, reliability structure.

1 Introduction

The questions of reliability traditionally concern problems connected with functioning of technical objects, and this term is very rarely used in relation to economic systems. In the economics literature, there is a considerable interest in the subject of risk. Since, according to the systems theory, the term "system" can refer both to technical and economic objects, it seems to be justified to transpose the general reliability theory to the sphere of economics and its use in risk planning and evaluation.

The general reliability theory defines reliability of an object differently from the classical theory. "A reliable object is an object, which functions in accordance with user's intentions, while an unreliable object is each object, which functions inconsistently with user's intentions" [1]. Specificity of production systems and, in particular, their complexity, allows treating them as operation systems, and then the reliability is one of their features measured by the extent of realization of determined indicators, parameters and characteristics. The most frequently analysed indicators of a production process include [2]: duration (t), efficiency (W) and productivity (P).

Transposition of the general reliability theory to the sphere of production systems can take place by treating unreliability (Z) – the opposite of reliability – as a synonym of risk (R) [1]:

$$R = Z\,. \tag{1}$$

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 389–396, 2011.

For this interpretation, the following equation should be true:

$$N + Z = 1. \tag{2}$$

This equation means that the probability that the system is in the state of reliability or unreliability is 1. In the face of the above, the following equations are also true:

$$N + R = 1, \text{ and hence } R = 1 - N. \tag{3}$$

Analysis and evaluation of risk will allow determining reliability of system functioning and vice versa. Despite the fact that the reliability approach in risk planning and evaluation offers more possibilities, it does not locate the risk factors in the system. Analysing the structure of a production system in the context of its reliability structure may provide a solution to this problem.

2 Reliability Structure of Systems

The structure of the system, which determines the relation between the state of reliability of the system and the state of reliability of its objects [1]. The analysis of the reliability structure of a system should be preceded by dividing the system into individual components – the system decomposition, which should reflect the logical connections in the system. In this paper will be presented only a parallel structure of production . Serial structures have been described, inter alia, in [3], [4].

According to the definition of the reliability of a system with a parallel structure says that the system is fit for operation, if at least one of its objects is fit for operation [1]. In the production practice, there occur parallel structures, however the nature of a production process does not allow for such interpretation of the reliable structure. The classical theory of reliability considers 0/1 states of technical equipment. This means that a production system would be recognized as reliable, if at least one element functioned correctly. In production systems, such a situation occurs only in so-called redundant systems, i.e. with a surplus of elements functioning in the system. In reality, redundant systems occur very rarely, because surplus of elements (e.g. machines, workers) means unused resources, which increased costs. Because production systems with redundancy occur in the production practice very rarely, a different way of interpreting and determining the risk for so-called parallel structures of production was proposed in this study. An example of the parallel structure of production can be the structure of the production system presented in Fig. 1.

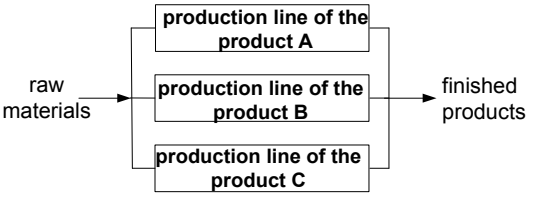


Fig. 1. An example of the parallel structure of production

For the n-element structure of the production system shown in Fig. 1, the risk of unreliability of one element R*i* should increase the total risk R_C^{PSR} of the system by the value R*i*. S*o*, the total risk should be the sum of risks of individual system elements.

$$R_C^{PSR} = R_1 + R_2 + ... + R_n = \sum_{i=1}^{n} R_i \,. \tag{4}$$

where R_1, R_2, R_n - the risk occurring in individual objects/subsystems of the system. If $\sum R_i > 1$, then $R_1 = R_1 / R_C^{PSR}$, $R_2 = R_2 / R_C^{PSR}$, $R_n = R_n / R_C^{PSR}$.

Individual risks R_i for n areas, depending on the amount of losses S_i incurred in these areas, will be as follows:

$$R_1 = \frac{S_1}{W_{teoret}}, \quad R_2 = \frac{S_2}{W_{teoret}}, \quad R_n = \frac{S_n}{W_{teoret}} \,. \tag{5}$$

where S_i – means a loss at the area *i* caused by occurrence of the risk factor r_i.

W_{teoret} – maximum value a selected indicator, which can be attained in theory (for example, theoretical capacity of a machine).

If the areas differ from each other, it is necessary, in case of such a type of structure, to determine the maximum value of the indicator attainable in the analysed technical system - W_{teoret} for each area. When determining the values of W_{teoret} for each of the *n* examined areas, individual losses S_i in these areas, depending on the time losses caused by occurrence of risk factors in individual areas, will be as follows:

$$S_1 = W_{teoret}^1 \frac{\Delta t_1}{T}, \ldots S_2 = W_{teoret}^2 \frac{\Delta t_2}{T}, \ldots S_n = W_{teoret}^n \frac{\Delta t_n}{T} \,. \tag{6}$$

where: W_{teoret}^i - means theoretical value of an indicator in individual areas of the decomposed system.

So, the total risk R_C^{PSR} for a system with n areas and parallel structure of production will be as follows:

$$R_C^{PSR} = \frac{W_{teoret}^1 \Delta t_1 + W_{teoret}^2 \Delta t_2 + ... + W_{teoret}^n \Delta t_n}{W_{teoret} T} \,. \tag{7}$$

If examined areas of the system are identical and are characterized by the same value of W_{teoret}^i, that is:

$$W_{teoret}^1 = W_{teoret}^2 = ... = W_{teoret}^n = \frac{W_{teoret}}{n} \,. \tag{8}$$

then the formula for the total risk of such a system will take the following form:

$$R_C^{PSR} = \frac{1}{nT} \sum_{i=1}^{n} \Delta t_i \,. \tag{9}$$

In accordance with the formula (4), the formula for the risk of the system from Fig. 1, will be as follows:

$$R_C^{PSR} = R_{lpwA} + R_{lpwB} + R_{lpwC} = \sum_{1}^{3} R_{lpwi} \ . \tag{10}$$

where l_{pwA}, l_{pwB}, l_{pwC} – individual production lines of a product.

3 Risk Acceptance Coefficient

At given organisational, technical and technological conditions the amount of risk of a given production system is constant. In case, when in one of areas/subsystems of a system the risk level is too high (unacceptable), one can try to lower the level of risk in that area, being denoted by i. Then however, risk levels in other areas/subsystems might alter, what is followed by necessary technological changes in other areas.

Change in level of risk - in individual areas/subsystems of a production system – down to a acceptable level can take place in a proportional or weight way.

3.1 Proportional Coefficient of Risk Changes

When using the proportional coefficient of risk changes, first and foremost one has to determine the level of risk, which is going to be acceptable in a given area/subsystem and where $R_i > R_{i\_akcept}$.

Proportional coefficient of risk changes WP will be determined as follows:

$$WP = R\_zm \frac{1}{n-1} \ . \tag{11}$$

where $R_{\_zm}$ – value of risk change within an entire system, n – number of areas/subsystems in a production system.

After taking into account the proportional coefficient of WP changes, risks in individual areas/subsystems will present as follows:

$$\begin{gathered} R_1 = \frac{S_1}{W_{teoret}}(WP+1) \ , \ R_{n-1} = WP\frac{S_2}{W_{teoret}}(WP+1) \ ,..., \ R_{n+1} = \frac{S_2}{W_{teoret}}(WP+1), \\ R_n = WP\frac{S_n}{W_{teoret}}(WP+1) \ . \end{gathered} \tag{12}$$

3.2 Weight Coefficient of Risk Changes

When using the weight coefficient of risk changes one can assume various alterations in risk levels in particular k areas, where $k \neq i$. The value of coefficients of changes in individual areas needs to satisfy the expression:

$$R\_zm = WP_1 + WP_{n-1},...,WP_{n+1} + WP_n \ . \tag{13}$$

After taking into account the weight coefficient for every area, risks in individual areas/subsystems will present as follows:

$$R_1 = \frac{S_1}{W_{teoret}}(WP_1 + 1),\ R_{n-1} = \frac{S_2}{W_{teoret}}(WP_{n-1} + 1)\ \textbf{,...,}$$
$$R_{n+1} = \frac{S_2}{W_{teoret}}(WP_{n+1} + 1),\ R_n = WP\frac{S_n}{W_{teoret}}(WP_n + 1)\ . \tag{14}$$

4 Example of Determining the Value of Risk for a Production System with a Parallel Structure of Production

The project was carried out in the Wrocław division of an international corporation. This Division is the third largest division within the corporation and it deals with production of electric locomotives, freight cars and subway cars.

The necessity of analysing production processes of 11 different products was a considerable difficulty. However, as both the production technology and organization of all production processes in the plant are similar, it was decided to perform the analysis for a representative product and to apply so called conversion factors in relation to other products. These factors determine the constructional and technological similarity in relation to the design, which is known and is deemed to be representative. As all production processes in the company are similar to the process of the representative product, it has been assumed that the same risk factors occur also in other processes.

4.1 A Method of Determining the Value of Risk for a Production System with a Parallel Structure of Production

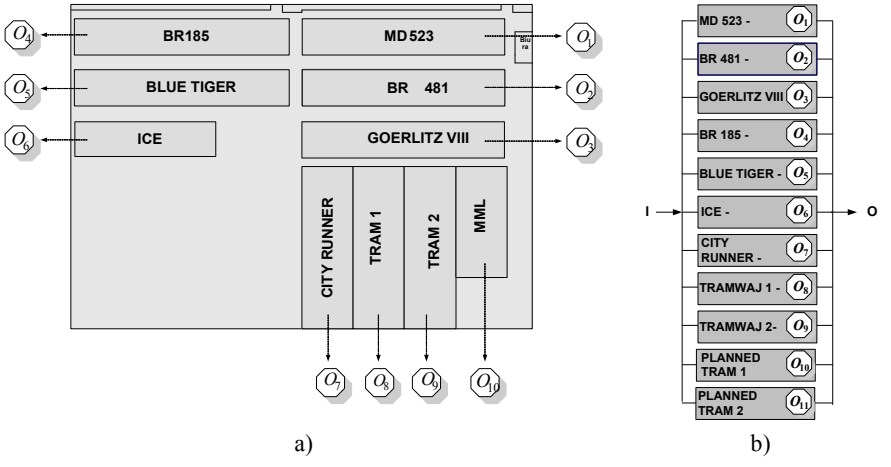
For functioning of the production system in the whole plant, functioning of all its areas is not a necessary condition. However, correct functioning of one area cannot be regarded as correct functioning of the whole system. The risk of unreliability of one area should translate into an increase in the risk of unreliability of the whole system, exactly by the value of the risk in this area. Therefore, it has been established that the reliability structure of the whole plant will be a parallel structure of production (presented in Fig. 2b).

So, when setting a goal for a production system, there should be considered the theoretically attainable production capacity, and not the quantities resulting from the sales plan. For production systems, the theoretical value is limited only by technological capabilities [5].

Therefore, taking into account only the times of production and transport, under the assumption that the system is fully reliable, resources are fully available, and the load of production lines is 100%, the theoretically attainable production capacity (W_{teoret}) for individual lines was set as the goal and was presented in table 1.

Since risk factors are of random nature and the period of observations provided a representative set of data, this period was accepted as representative for determining

characteristics of risk factors [6]. The annual pool of working time for the plant is 46 weeks, for T=3 months= 12 weeks. Table 3 presents the data needed for determining the total risk R_C^{PSR} of the whole plant.



a) b)

Fig. 2a). Designations accepted for areas of the production system of the whole plant, **b)** parallel structure of production assumed for the analysed plant

Table 1. Data required in the method

Project name	Conversion factor	W_{teoret} / T	R_i	Δt_i taking into account the conversion factor and the quantity planned for production
BR 185	1.7	35	R_4	5.17
Blue Tiger	1.9	17	R_5	2.81
ICE	1.6	52	R_6	7.23
MD 523	1	138	R_1	12
Goerlitz VIII	1.4	104	R_3	12.66
BR 481	1.2	69	R_2	7.20
Cityrunner, Tram 1, Tram 2	1,1	35	R_7, R_8, R_9	3,35
Planned 1, Planned 2	0.4	35	R_{10}, R_{11}	1.22

Using the formula (7) values of losses in individual areas, which are equivalent to products, were determined. Knowing the value of losses from the formula (6) the risks for the examined areas can be determined.

It was assumed that the level of risk acceptance for the MD 523 product equals 0.1. By means of proportional coefficient of risk changes based on the formula (11), levels of risks were found in individual areas of the analysed system. Additionally for individual are of the production system weight coefficients were assumed and based on the formula (12) for weight coefficient of risk changes, levels of risk were

determined for individual areas of the analysed production system. Obtained in that manner values are presented collectively in the table 2.

Table 2. Values of losses and risks for individual products

Project name	Area design ation	Losses [pcs/12 weeks]	Risks	Level of risks after applying WP	Values of assumed weight coefficients	Levels of risks after applying *R_zm*
BR 185	O4	10.61	0.018	0,020	0	0,018
Blue Tiger	O5	2.92	0.005	0,006	0,1	0,006
ICE	O6	22.35	0.038	0,041	0,1	0,041
MD 523	**O1**	**99.1**	**0.168**	**0,100**	**-**	**0,100**
Goerlitz	O3	78.23	0.133	0,138	0	0,133
BR 481	O2	29.73	0.051	0,054	0,1	0,054
Cityrunner	O7	6.86	0.012	0,013	0,1	0,013
Tram 1	O8	6.86	0.012	0,013	0,1	0,013
Tram 2	O9	6.86	0.012	0,013	0,2	0,015
Planned 1	O10	2.5	0.004	0,006	0,1	0,006
Planned 2	O11	2.5	0.004	0,006	0,2	0,008

Obtained levels of risk before and after application of coefficients of risk acceptance are combined together in Fig.3.

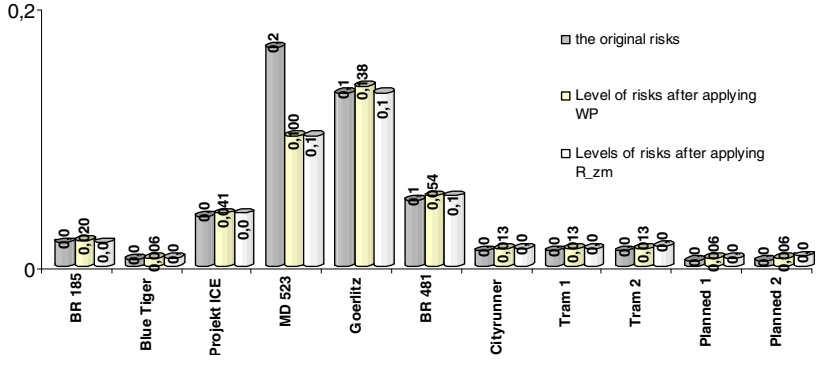


Fig. 3. Value of risk for individual products

When calculating the total risk for the whole plant R_C^{PSR}, the formula (4) should be used. Then the total risk will be:

$$R_C^{PSR} = R_1 + R_2 + ... + R_{11} = \sum_{i=1}^{n} R_i = 0{,}46 .$$

The level of the risk of the entire plant in relation to the risk of the representative product decreased.

5 Conclusion

The production system was defined in accordance with systems theory, while the risk was treated as a synonym of unreliability. This approach allowed decomposing the production system into several areas and determining the reliability structure of the production system. This paper presents a method of determining the risk for a production system with a parallel structure. Additionally weight and proportional coefficient of changes' risk was introduced. The method of determining the risk for a parallel structure of production was verified in a production company, which manufactures bogie frames for railway cars, trams and railway engines.

References

1. Bizon Górecka, J.: Engineering of reliability and risk in enterprise management. Printing House of the Organisational Development Centre, Bydgoszcz (2001)
2. Bunse, K., Sachs, J., Vodicka, M.: Evaluating Energy Efficiency Improvements in Manufacturing Processes. IFIP Advances in Information and Communication Technology 338, 19–26 (2010)
3. Chlebus, E., Burduk, A.: Modelling and variant simulation in risk evaluation of starting-up production programme. In: 1st International Conference on Changeable, Reconfigurable and Virtual Production, Agile (2005)
4. Mlynczak, M., Nowakowski, T.: Rank Reliability Assessment of the Technical Object at Early Design Stage with Limited Operational Data A Case Study. International Journal of Automation and Computing 1, 169–176 (2006)
5. Cichosz, P.: Economic aspects of cutting with diamond-coated wire. Archives of Civil and Mechanical Engineering 4, 5–14 (2008)
6. Kobylinski, L.: System and risk approach to ship safety, with special emphasis of stability. Archives of Civil and Mechanical Engineering 4, 97–106 (2007)

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

Damian Krenczyk and Bozena Skolud

Silesian University of Technology, Institute of Engineering Processes
Automation and Integrated Manufacturing Systems,
Konarskiego 18A, 44-100 Gliwice, Poland
{damian.krenczyk, bozena.skolud}@polsl.pl

Abstract. Fast growth of the SME sector, increasing demands of customers and dynamic market force the producers to lower production costs and to use new IT tools in production. The paper presents method of integration of preparation of production and production planning systems, such as SWZ and PROEDIMS. The system constituted in this process will enable SME entrepreneurs to take correct decisions connected with planning and controlling production. The integration will be achieved by methods of data transformation and data mapping using XML language. Due to this integration PROEDIMS system will be enriched with the module supporting verification of production orders, using constraints satisfaction and depth-first search (DFS) algorithm with backtracking.

Keywords: production planning, integration, xml, constraint satisfaction, organisation variants.

1 Introduction

Nowadays, production enterprises operate in a very dynamic environment. Global market, fierce competition and constantly changing customers' demand lead to shortening product life cycles, together with increasing the complexity of products. These factors are forcing manufacturers to adapt to new circumstances and to invest in increasingly sophisticated and innovative technologies. Along with these changes there is a need to develop and implement new production systems planning methods [1, 2]. These demands concern mainly small and medium-sized enterprises (SME), which in the EU constitute 99.8% of all non-financial business economy enterprises, which represents almost 70% of total employment in the private sector. SME companies are divided into medium-size (fewer than 250 employees), small (fewer than 50 employees) and micro-enterprises (employing fewer than 10 people). In the manufacturing sector, SMEs constitute 99.2% of enterprises. Micro-enterprises, which currently account for 92% in the SME are undergoing especially high growth in recent times [3].

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 397–404, 2011.

For SMEs segment companies, it becomes an imperative to use of computer aided decision support systems in production planning process, in particular with regard to decisions on the possibility of production order implementation which guarantees realization of the production job according to the production order, with the lowest level of capital invested. Support systems in SME sector functioning in various areas related to the preparation and production planning are usually not integrated. Lack of integration between these systems affects the efficiency in their use and is a potential area where it is possible to increase efficiency, which is associated with reduction of costs for SMEs.

2 Problem Formulation

Because of the wide range of activities, high cost of implementation and a need to apply changes within a company's structure, production management systems available on the market are implemented mainly in big enterprises. For SMEs (and particularly for micro companies) the cost and time of implementation of MRPII / ERP class systems are main reasons due to which they are not implemented. The situation of SME presented in the introduction, connected with rapid development, execution of multi-assortment production realized in small batches which is often a reaction to the sudden demand (MTO), force companies to look for tools to support decision making at the operational level and orders management.

The necessity to build IT tools involves a need to develop new decision-making support methods, in particular with regard to decisions on the possibility of the new production order acceptance, realization of the production job according to the production order, with the least cost of production. In the paper results of the implementation phase of the exchanges data module are presented. The exchanges data module is necessary in the process of integration of production preparation module of PROEDIMS system with production orders verification SWZ system for multi-assortment, concurrent production. The constraint satisfaction techniques are used in SWZ system. Sufficient conditions for all possible solutions of production flow filtering are defined using this approach and it gives a set of admissible solutions for both the customer and the producer demands.

Meeting this goal will require modifications to independent systems operated in areas related to the preparation and production planning. PROEDIMS system is being developed at the Institute of Machine Technology and Automation at the University of Technology in Wroclaw. SWZ system is being developed at the Technical University of Silesia. Integration of these systems will be implemented through the development and implementation of a dedicated interface for data exchange. In the paper the method of formal description of production processes data structures taking into account available resources of production system using Extensible Markup Language (XML) is presented. The exchanging data interface module based on techniques of data transformation and data mapping supporting the integration of computer applications is also presented.

3 SWZ and PROEDIMS Production Planning and Preparation Systems

3.1 PROEDIMS System

PROEDIMS system belongs to a family of products which manage product data and processes in the enterprise. PROEDIMS enables creation, collection, management and propagation of all data related to the product throughout the product life cycle and all information and data necessary for the proper functioning of the company.

This system supports various areas and activities related to the product and company activities, starting from the conceptual phase, throughout design, process management, logistics, customer and suppliers relationships management, to the maintenance and servicing of products [4]. Phase of the project discussed in this paper, is related to the integration of the SWZ system with PROEDIMS, and intends to complement the PROEDIMS with the module associated with the jobs scheduling on system resources. Input data of SWZ system transferred from PROEDIMS system allows determination of the sequence of operations on the resources of the production system. After transferring it back into PROEDIMS, the data will become the basis to generate production system work schedule.

3.2 SWZ System

Production Orders Verification System – SWZ is a computer implementation of methods supporting rapid decision taking on the acceptability of a production order for multi-assortment repetitive production systems. The current state of knowledge and achievements in the area of scheduling significantly limit their practical application. Determination of the optimal schedule, in most cases is a NP-hard problem (NP-complete). Algorithms with polynomial computational complexity for NP-hard problems probably do not exist, which practically means that for most manufacturing systems it is not possible to achieve the optimal solution in reasonable time, due to the number of resources and jobs. This necessitates the abandonment of determining the set of all possible solutions for the determination of a subset of feasible solutions. In real production systems it is required not only to find the best possible solution, but to find a solution possible to reach in a reasonable time (a solution acceptable) and the one that can be accepted given the existing constraints. The group of methods that guarantees obtaining an acceptable solution includes the artificial intelligence methods, heuristics and meta heuristics, random search methods, simulation methods or constraints satisfaction [5].

The support rapid decision-making methodology on the acceptability of a production order using constraints satisfaction techniques and is tantamount to testing a sequence of arbitrarily selected conditions. The fulfilment of all conditions (their conjunction) guarantees the possibility of order execution (Fig. 1). Lack of solution provides information about the necessary abandonment of specified conditions of the order, or having to meet the needs associated with an increase in available capacity, storage space, etc.

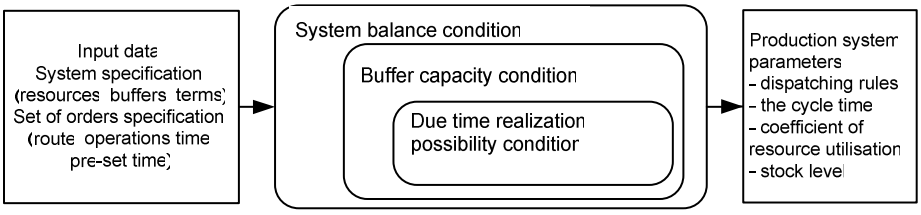


Fig. 1. Procedure of the acceptance of the production orders set for realisation in the system

Sufficient conditions, which are used in described methodology, have been designated for the production system and production order identified constraints. The conditions include: [1, 6, 7]:

- system balance condition - takes place when the number of processes introduced into the system is equal to the number of processes leaving that system during one system cycle,
- buffer capacity condition - the capacity of the inter-resources buffer is equal or bigger than the realization number of the process during one system cycle.
- due time realization possibility condition – processes included in the production order will be executed within the due time required by the customer.

In previous works [1, 2, 8] it has been proved that for multi-assortment repetitive production (for the steady state of system) fulfilment of the system balance condition and the buffer capacity condition provides a qualitative functioning of the system (deadlock-free behaviour of a system). However, the growing market requirements for increasingly varied products cause that production is realized in short series, and thus the process flows are often changed. A significant problem is therefore a transition from one expected steady state of the production system to another one.

In the considered methodology the deadlock protection method (which guarantees functioning of the system for transient phases) is used and it establishes the sequential ordering of the processes realisation in the dispatching rule (for the start-up and cease phase). Start-up rule realizations fill up the additional number of elements into the inter-resources buffers, which guarantees deadlock-free system behaviour and synchronises the production flow into the expected steady state. Similarly the cease phase applies to the single process completion or final production completion. The transient phase includes the starting-up phase and cease phase as well. The transient phase consists in the transition from one expected steady state of the system to another one [9]. During the starting-up/cease phase the starting-up/cease rules are determined.

The determination of the starting-up/cease rules consists of the following stages:

- the identification of close cycles in the system structure,
- the determination of the multiplication of the process realized during the starting-up/cease rule.
- the determination of the processes realization sequence in the starting-up/cease rule.

The cycle of mutual expectations is one of the conditions necessary for the deadlock occurrence. The equivalent to ensuring deadlock-free system behaviour is avoiding the appearance of that cycle through the allocation of the dispatching rules at the resources. According to the production system topology, the cycle of mutual expectations can take place only at the shared resources belonging to the closed cycles. Thus, at first the closed cycles should be identified. For the identification of the close cycles occurring in the production system the contour search algorithm adopted from Graph Theory is used [8]. It is possible because a graph is a model that enables the description of a production system structure. The production resources are represented by the graph vertices and the parts of the production routes allocated between each pair of the neighbouring resources are represented by the ordered pair corresponding with the graph edges. The elementary contour from the terminology of Graph Theory is adequate to the basic cycle from the terminology describing the production system structure. The identification of the graph contours requires the application the deadlock protection method. That method requires the action consistent with the consecutive steps of the contour graph algorithm including the depth-first search (DFS) algorithm with backtracking. According to the DFS algorithm one starts from the start vertex and moves along the path generated by the successors list. The successors of checked vertex create the tree of next successors. The consecutive vertices are checked until it hits a vertex that has no non-checked successors. Then the search backtracks, returning to the most recent vertex it hasn't finished exploring. When all routes starting from the x vertex are verified it is necessary to choose another vertex and repeat all procedures. It means that the recurrent procedure of the graph contours search is considered. The deadlock protection method enables to check, whether the contours in the given graph exist and find the sets of the graph vertices forming the contours. The pseudocode of the contour search algorithm presented in Fig. 2 can be found at [10].

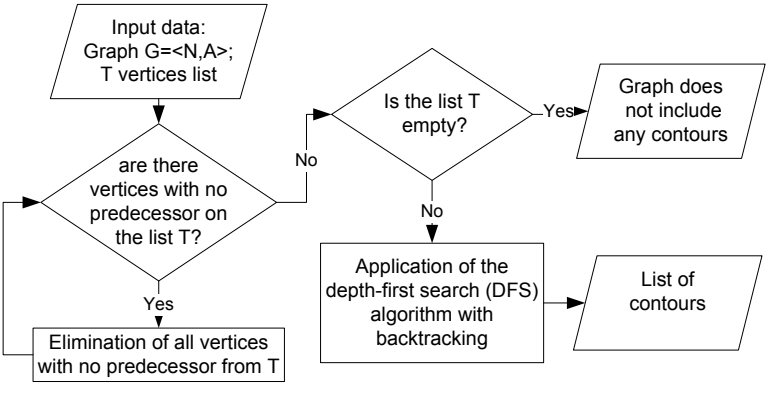


Fig. 2. Contour search algorithm

The result of that stage is the list of the resources belonging to the basic cycles. If the list of resources is empty, the determination of the processes realisation sequence in the starting-up rule is useless. The support rapid decision-making methodology on the acceptability of a production order which uses elements of constraints propagation

method and DFS algorithm has been implemented in the SWZ. For the data describing the system and the production order dispatching rules are determined, together with quantitative and qualitative indicators of the production system. It gives the possibility to form high-level indicators of production, such as resource utilisation and level of work in progress.

4 SWZ and PROEDIMS Exchange Data Module

The exchange of data between systems, due to the versatility and convenience in use, is achieved using the Extensible Markup Language XML [11]. XML is currently very popular and much more often used in the exchange and analysis of data collected and processed in IT systems, supporting enterprise management at different levels and functional areas. XML is designed to represent different data types in a structured way. The choice of XML for integration PROEDIMS and SWZ systems, as the language for collecting data, was dictated by the fact that XML is currently very popular and used more often in areas related to the exchange and analysis of data in enterprise management systems [12, 13, 14]. For production system and production order models developed the document structure definition using XML Schema. The choice was dictated by the way of writing the definition, which is also implemented using XML, and the fact that XML Schema allows to define data type constraints. It also allows to create new definitions of the structure or combining information from several schemes, which is important in the process of acquiring data from enterprise resource planning systems.

Developed for integration needs, XML schema in PROEDIMS system has been divided into the following modules:

- Planning - containing data on production orders and operations for scheduling.
- Production - containing data on orders and related operations currently in progress,
- Resources - containing a list of production resources with the calendars of availability.

Developed XML schema for SWZ system defines the structure of an document describing the production resources consisting of manufacturing system, production processes data and production flow data.

The next step in the implementation of a systems integration project was creation of a transformation data module between different data models. The transformation process was divided into two concurrent stages resulting from the functional areas of integrated systems:

- The stage of calculation, associated with the change of the generated sequence of the tasks on the resources of the schedule used by the system PROEDIMS. This stage is related to the fact that the system implementation schedule PROEDIMS operations on resources is associated with the calendar availability of productive resources, which is not taken into account by the SWZ system.
- The stage of data mapping stored in PROEDIMS data model and data set in the previous stage with the data which is stored in SWZ data model.

For the purpose of transformation process automation was used Extensible Stylesheet Language Transformations XSLT [15]. XSLT is used to convert an XML document into another document, Web page, a text document or other file type. Implementation of the conversion process can be performed using the processor delivered as standalone products, or as components of other software including web browsers, application servers or other open-source software. Calculations on the data are realized with the use of XML Path Language.

Data module was implemented in SWZ system. Functional diagram of a transformation module was shown in Fig 3. Implementation of the transformation data module in SWZ system consists of the following steps:

- loading the XML file, containing information on the planned production, from the PROEDIMS,
- validation of the loaded file based on XML Schema,
- transformation of the file using XSLT (mapping data and calculations using XPatch),
- generating an XML file for SWZ,
- validation of the loaded file based on XML Schema.

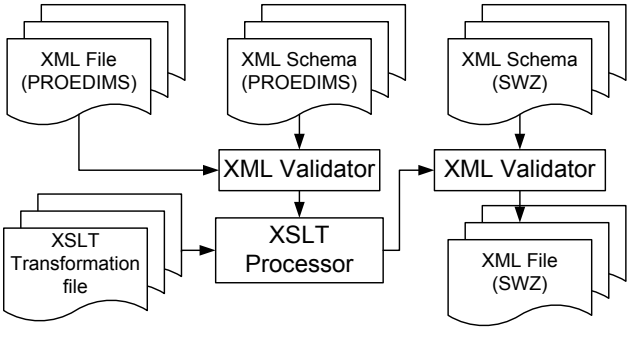


Fig. 3. Transformation data module

Elements of the structure of input and output files are converted according to an order resulting from the requirements of an XML document formats through properly addressed references to the tags (nodes) using the XPatch language. In the same way the next phase will be conducted involving the integration of processing and transmitting information containing the schedule generated from SWZ to PROEDIMS system.

5 Summary

Presented in the paper method of exchanging data between different preparation and planning of production systems, based on data transformation and data mapping allows integration of the systems described in the paper. The proposed integration module will increase the efficiency of planning departments and performance of production systems, which is associated with reduction of costs for SME. As a result

of the development and implementation of the system it will be possible to increase the effectiveness of the integrated decision-making areas. It will also allow SME sector companies to create virtual organizations. The prototype of the management system dedicated for SMEs will be the final result of the project.

Acknowledgments. This work was supported by The National Centre for Research and Development as a part of a No. N R03 0073 06/2009 research project.

References

1. Skolud, B., Krenczyk, D.: Flow Synchronisation of the Production Systems – The Distributed Control Approach. In: 6th IFAC Workshop on Intelligent Manufacturing Systems IMS 2001, Poznan, Poland, pp. 127–132 (2001)
2. Skolud, B., Banaszak, Z.A.: Modelling of distributed control for repetitive production flow prototyping. International Journal of Computer Integrated Manufacturing 18, 386–394 (2005)
3. European Commission, Annual report on EU small and medium-sized enterprises (2009), http://ec.europa.eu/enterprise/policies/sme/facts-figures-analysis/performance-review/pdf/dgentr_annual_report2010_100511.pdf
4. Cholewa, M., Czajka, J., Konopa, A.: CAD and internet-based enterprise resource planning systems integration (in Polish). Mechanical Review (Przeglad Mechaniczny) 3, 9–12 (2009)
5. Kalinowski, K.: Scheduling of production orders with assembly operations and alternatives. In: Nabhani, F. (ed.) 19th International Conference Flexible Automation and Intelligent Manufacturing, p. 85. Curran Associates, Inc., Middlesbrough (2009)
6. Banaszak, Z., Skolud, B., Zaremba, M.B.: Computer-aided prototyping of production flows for a virtual enterprise. Journal of Intelligent Manufacturing 14, 83–106 (2003)
7. Salido, M.A.: Introduction to planning, scheduling and constraint satisfaction. Journal of Intelligent Manufacturing 21, 1–4 (2010)
8. Krenczyk, D., Dobrzanska-Danikiewicz, A.: The deadlock protection method used in the production systems. Journal of Materials Processing Technology 164, 1388–1394 (2005)
9. Dobrzanska-Danikiewicz, A., Krenczyk, D.: The method of the production flow synchronisation using the meta-rule conception. Journal of Materials Processing Technology 164, 1301–1308 (2005)
10. Pseudocode of the contours search algorithm implemented in the SWZ system., http://imms.home.pl/swz/contour.html
11. Extensible Markup Language (XML) 1.0 (5th edn.) W3C Recommendation, http://www.w3.org/TR/2008/REC-xml-20081126/
12. Tao, Y.H., Hong, T.P., Sun, S.I.: An XML implementation process model for enterprise applications. Computers in Industry 55, 181–196 (2004)
13. Wang, C.G., Xu, L.D.: Parameter mapping and data transformation for engineering application integration. Information Systems Frontiers 10, 589–600 (2008)
14. Lee, C.K.M., Ho, G.T.S., Lau, H.C.W., Yu, K.M.: A dynamic information schema for supporting product lifecycle management. Expert Systems with Applications 31, 30–40 (2006)
15. XSL Transformations (XSLT) Version 2.0, W3C Recommendation (2007), http://www.w3.org/TR/xslt20/

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

Cezary Grabowik and Krzysztof Kalinowski

The Silesian University of Technology, The Faculty of Mechanical Engineering,
Institute of Engineering Processes Automation and Integrated Manufacturing Systems
Konarskiego 18a Street, 44100 Gliwice, Poland
{Cezary.Grabowik,Krzysztof.Kalinowski}@polsl.pl

Abstract. In this paper some aspects of CAD/CAPP/CAP systems integration process are described. Moreover, the structure of an integrated environment, which is an answer to this kind of integration problem is shown. Main elements of this environment are technological knowledge base – TKB and scheduling knowledge base – SKB. The foundation for building these knowledge bases was application of object methods in processes of technological and scheduling knowledge representation. In the paper both models it means the object model of technological knowledge representation and scheduling knowledge representation are discussed in details.

Keywords: CAD, CAPP, CAP, process planning, multi variant processes, scheduling, object methods, disturbance, knowledge representation.

1 Introduction

The complexity and actions specificity of the technical and organizational planning and production control domain motivate to the practical application of artificial intelligence tools in an integration process of CAD/CAPP/CAP systems. Proposed by the authors the integration methodology of a computer aided design, technological and organisational production preparation differs from the others integration methods because it can perform the full integration mentioned above domains of engineering activities. The most important reason for undertaking researches on the integration in these fields of engineering activities is to make a production system more efficient and flexible. The increasing of a production system effectiveness can be achieved by means of the information integration that is processed by designing, technological and production planning company divisions [1,2]. Apart from the methodology used in order to integrate processes of a product designing, process planning and production planning the following question appears: who should be a user of this integrated environment? Should it be a designer or a process engineer? These questions make us aware of a fact, that in former researches, which treat about integration problems, there is not any information about the origin of technological process plans, whilst the process plan is essential for integration CAPP and CAP systems. From the other hand ERP systems try to apply simulations techniques in order to solve decision problems at the shop floor level, but this solution is limited because of lack of time, usually it is

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 405–412, 2011.

not possible to get satisfying solution in a reasonable period of time. Moreover in the ERP systems the production schedule is usually calculated once a week it means that in most cases it is not possible to react to some disturbances which can appear between successive schedule calculations so their consequences are taken into account in the following planning period. Taking into consideration above it seems to be necessary to look for an alternative solution. During the author's researches the following assumptions were made:

- a CAD system output data have to be represented in the object manner apart from a CAD system internal product design representation,
- the technological knowledge have to be represented in object manner that is essential for assuring the inner coherence of the integrated environment,
- a CAPP system output data should be represented by set of multi variant processes. The multi variant process should be represented with xml file,
- a CAP scheduling rescheduling system should allow to calculate a production schedule either forward or backward direction depending on a user decision,
- a CAP system scheduling rescheduling knowledge base has to be organized in the object way,
- a CAP system together with the CAPP system has to be able to react to certain disturbances types,
- a CAPP and CAP systems should have an open structure, they should assure an interface of data exchange with commercially available ERP software.

Based on above cited premises the structure of the integrated CAD/CAPP/CAP environment was established. In this structure the following base elements can be distinguished (see: the figure 1): DFD – a design features database, TKB a technological knowledge base, TDB – a technological database, SKB – a scheduling/rescheduling knowledge base, PMD – a product model description, SMP – a set of multi variant process plans, PSH – a production schedule. As one can see in the figure 1 there is a design–technology feedback established between a CAD and CAPP systems and schedule–technology feedback set up between CAPP and CAP systems. In the traditional approach a technological process planning begins with a product design analysis. As a result of this analysis sometimes it is necessary to make some modifications in the product design. Thanks to the design–technology feedback the information what should be done with the product design can be easily transferred into a CAD system. The schedule–technology feedback is made based on a set of manufacturing alternatives. Multi variant technological processes are used as follows [1,2,3,4]:

- providing there is a disturbance in a production system so schedule is out of date. A CAP system sends information about the problem and reports which manufacturing process is stopped. Moreover the CAP system passes on information about the state of a product being manufactured, it means which features of the product are already done and undone;
- the CAPP system first checks whether it is possible to continue the technological process with one of available alternative technological routes, if not it reports that the product is a failure. If yes, the CAPP system sends information about the technological route, which has to be applied in order to finish the product and the CAP system calculates a new production schedule.

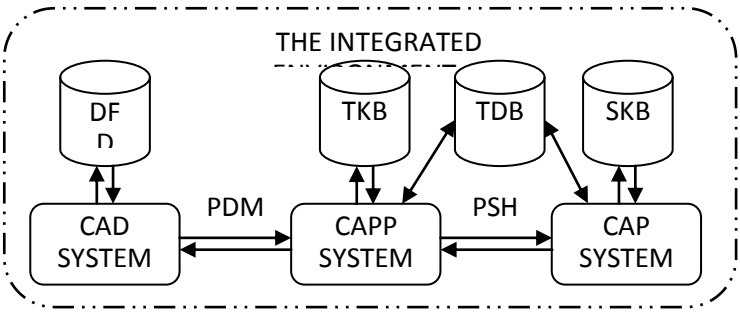


Fig. 1. The structure of the integrated CAD/CAPP/CAP environment

In order to build a knowledge base of an expert system the object-oriented methodology (OOT) was used. Application of the OOT methodology in a system building process is performed in three stages, they are as follows: object oriented analysis (OOA), object oriented design (OOD) and object oriented programming (OOP). The object-oriented analysis is applied in order to model a system problem domain. During this stage particular objects and classes are being searched, the initial hierarchical class structure is also being created. One of the most important and critical decision made during the object oriented design is a choice of a system implementation programming language. This so because the class structure achieved at the analysis stage in most case has to be adapted to programming language characteristic (inheritance mechanism). The object-oriented programming is the stage when the complete program is being programmed according to results of the object oriented analysis and design stages. Although currently the process of software development reached the programming stage this paper shows only results achieved during the object oriented analysis and design stages. This choice was made taking into consideration the importance of these stages for software quality and implementation costs.

2 The Object Method of Technological Knowledge Representation

As it was written in the previous paragraph, one of the main elements of the integrated environment is the technological knowledge base – TKB. The model of technological knowledge representation, based on an object paradigm, was the foundation for building the TKB. This model allows to represent knowledge from a domain of technological processes planning for axis symmetrical parts. The technological knowledge is represented by means of a hierarchical class structure because of defining in the inner structure of particular classes design rules. These design rules are responsible for planning of technological cuts and operations, cutting tools selection, inspection tools selection and additional technological instrumentation selection. The elaborated object model has declarative procedural character. The declarative character of proposed method arises from the fact of recording in particular methods designing rules. These rules are used for planning chosen elements of a technological process plan. The procedural character results from recording in a

content of particular methods SQL queries to suitable technological database tables. The base class in the object model is *TProcessPlan* class (Fig. 2). The main attribute of the *TProcessPlan* class is *TypeObTec* attribute used for distinguishing what kind of a structure element of a technological process plan a given instance of the *TProcessPlan* class represents. There are the two possibilities a given instance can represent either a technological operation or a technological cut. Taking into consideration fact that a given technological object, an instance of any class belonging to the domain of the *TProcessPlan* class, can be created according to the logic of a technological process planning for different design features introducing of the attribute *IdObDes* in the inner structure of the *TProcessPlan* class was necessary. This attribute represents an identifier of a design feature for which given instance, the technological object, was created. Moreover, the *TProcessPlan* class plays the role of some kind of container, which means that it collects certain set of attributes common for other classes belonging to its domain. These attributes represent for example information about manufacturing allowance values, manufacturing parameters etc.

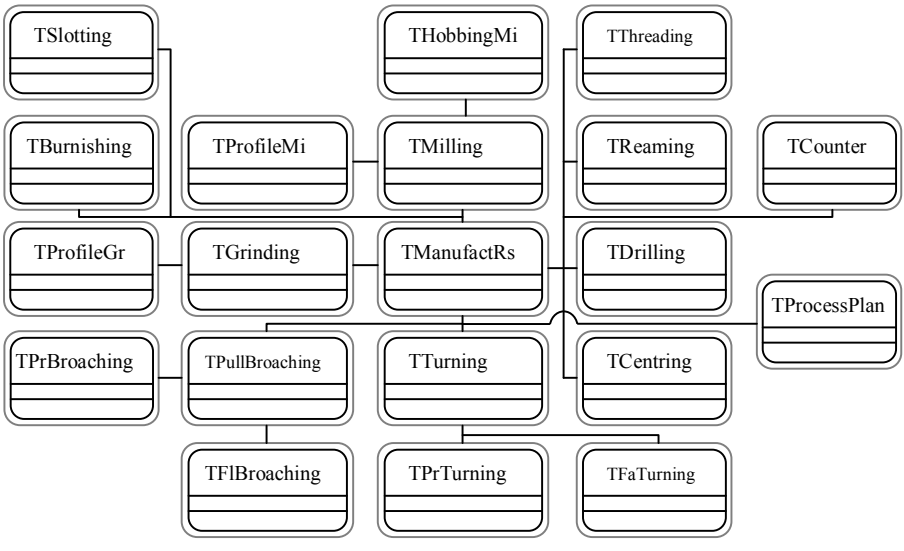


Fig. 2. The object representation of technological knowledge

The *TManufactRs* class is the class that directly inherits from *TProcessPlan* class. This class was introduced in the hierarchical class structure in order to collect common attributes that represent manufacturing resources such as: cutting tools, tool holders and technological instrumentation. This class is the base class for other classes in the hierarchical class structure. These classes represent particular manufacturing techniques. So, turning manufacturing technique is represented by the *TTurning* class. In the *TTurning* class structure the group of attributes that represent different manufacturing strategies for outer cylindrical surfaces machining were introduced. These attributes are as follows: *StratRgMach*, *StratShMach*, *StratFiMach*.

Thanks to introducing these attributes, it is possible to plan a technological process plan according to the following strategies: according to the longest cutting path, according to the shortest cutting path, according to a mixed strategy. This feature of the *TTurning* class makes a planning of multi variant processes possible. For the sake of the possibility of different manufacturing resources application (different insert shapes, different tool materials and different tool holders etc.) with regard to the kind of machining (straight turning, facing), considered process planning decision stage (a rough machining, a profiling, a finishing) in the inner structure of the *TTurning* class the attribute *SeqInSh* and *SeqHlSh* were introduced. These attributes are responsible for putting in order processes of insert and tool holder system choice. The *TTurning* class is the base class for the two classes called adequately the *TFaTurning* (facing) and *TPrTurning* (profiling). These classes contain attributes and class methods, which take into account the characteristic features of a turning technique with application of a cross-feed and a profiling machining. A centre hole machining technology is represented by the *TCentring* class. Holes machining methods in the model of technological knowledge representation were recorded with application of the following classes: *TDrilling* (holes drilling), *TReaming* (holes reaming), *TCounter* (counterboring, countersinking). The milling machining technique in the object class structure is represented by means of the *TMilling* class. The *TMilling* class is the base class for *TProfileMi* and *THobbingMi* classes. In the inner structure of the *TMilling* class only general information about milling process are kept. The *TProfileMi* and *THobbingMi* classes as specializations of the *TMilling* class take into consideration characteristic features of a profile and hobbing milling. The object model was supplemented with the *TSlotting* class that represents manufacturing techniques applying for manufacturing design features such as: slots and teeth. The TGrinding and TProfileGr classes represent grinding machining methods. The *TGrinding* class describes machining methods for flat surfaces whilst the *TProfileGr* class for profile shapes such as teeth, splines etc.

3 Object-Oriented Model for Production Scheduling

Another important part of presented integrated environment is a scheduling knowledge base – SKB. Its construction was based on object-oriented knowledge representation. Example object structures for scheduling domain were presented in [1,5]. The object-oriented analysis identifies subjects, classes and objects, structures, attributes and definition of methods and messages. In [6,7] methods of production scheduling and control domain was presented. The result of object-oriented analysis is a formal model of the proposed system (Fig. 3.).

3.1 Subjects

"Subject" in the object-oriented technology is a dedicated group of related objects. The subject may be interpreted as a model of some subsystem - the use of themes allows for a logical separation (clustering) of selected areas of the modelled domain. In the domain of production scheduling four subjects are identified:

- TPP – technological processes planning – refers to the structure of technological processes plans of all products, including the variants of operations.
- PRODUCTION AND PLANNING – it covers both production orders already planned and implemented in the production system.
- PRODUCTION SYSTEM – includes the structure of the production system – resources and calendars of their work.
- SCHEDULING – saves the dates at which resources are occupied by the tasks and breaks.

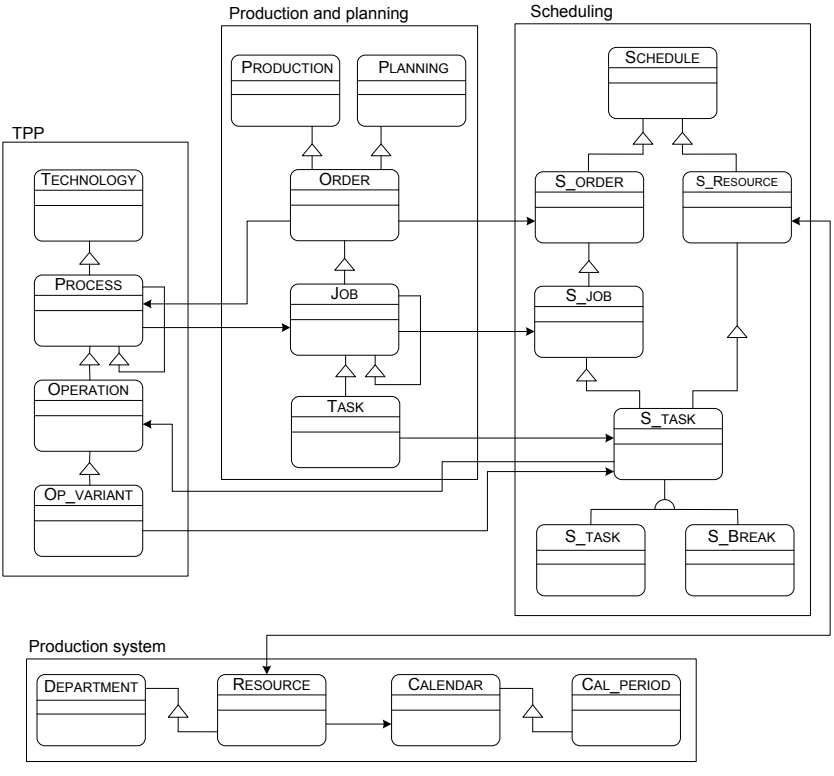


Fig. 3. Object-oriented analysis model of production scheduling. Structural connections.

3.2 Classes and Objects

In the TPP subject the main class *Technology* includes following classes: *Process*, *Operation* and *OP_Variant*. The *Process* class identifies technological processes plan of a product, if the product is composed of other parts, then it has references to its sub-components (tree structure). *Process* consists of the list of operations – *Operation* class refers to the main part of the process performed on a single resource. In many cases capabilities of a manufacturing system allows realizing an operation on more

than one resource. Such parallel resources can be identical or can work with different efficiencies (e.g. different operation times, setup times). Therefore, for each possible resource a separate *Op_Variant* is assigned.

The main classes in PRODUCTION AND PLANNING subject are *Production* and *Planning*. Class *Production* represents system occupancy – orders that are realized in the system and those confirmed for realization with fixed starting times of operations. Class *Planning* consist of orders for scheduling. Class *Order* represents production order and can be composed of one of more *Job*s in accordance with the technological process. Class *Task* represents activities related to the implementation of the particular operation.

In the PRODUCTION SYSTEM the main class is *Department*. It enables to model internal substructures in an enterprise. Instances of the *Department* class include all the *Resource*s (machines, work places) in which tasks are performed. Each *Resource* has a fixed calendar time – established by instances of *Calendar* and *Period* classes.

The main class in SCHEDULING subject is *Schedule*, which represents activities on resources in given period. *Schedule* consists of *S_Order*, *S_Schedule*, *S_Job* and *S_Task* classes.

3.3 Structures

The structure layer defines relations between classes and their instances in the object model. Three kinds of relations were distinguished:

- generalization/specialization – e.g. *S_Period/S_Task* (edges with semi-circles),
- aggregation/decomposition – e.g. *Process/Operation* (edges with triangles),
- transmitter/receiver – e.g. (directed arcs).

3.4 Attributes

Attributes layer includes both the attributes of objects and relations between objects (relations between instances). The attributes of the object are the data of various types. For example, object of class *Order* is characterized by the attributes as “name”, “priority”, “batch size”, “lot size”, “release date”, “due date”, “main process”, “scheduling strategy”, “lot flow” etc. Relations between instances show how the object in the class must be associated with an object(s) in another class. This procedures also define the size (1:1, 1: M, M: M - one to one, one to many, many to many) and participation (optional or required) for each relation.

In the figure 3 the subjects, the aggregation/decomposition and the generalization/ specialization structures and the relations between instances of the object-oriented analysis model are presented.

3.5 Methods and Messages

Methods and messages layer, opposed to the previous layers, defines dynamic relations between objects. The method is a specific action of the object, which is required to perform by itself. Messages trigger execution of specific tasks using specific methods of the object. Example flow of messages is shown in Fig. 3.

4 Summary

Object-oriented applications development distinguishes three phases: analysis, design and programming. In the paper, the analysis phase of the object-oriented models of the technological processes preparation and the scheduling were presented.

Computer aided production scheduling can dramatically reduce the time-consuming search for a solution, especially in systems with large numbers of resources and processes. Presented methodology is focused on supporting effective decision making in the verification of production orders and optimizing the production flow. Elaborated structures allow the development of dedicated computer systems with a knowledge base (expert systems).

Acknowledgements. This work has been conducted as a part of research project N R03 0073 06/2009 supported by The Polish National Centre for Research and Development (NCBiR).

References

1. Grabowik, C., Kalinowski, K., Monica, Z.: Integration of the CAD/CAPP/PPC. Journal of Materials Processing Technology 164–165, 1358–1368 (2005)
2. Grabowik, C., Knosala, R.: The method of knowledge representation for a CAPP system. Journal of Materials Processing Technology 133, 90–98 (2003)
3. Sormaz, D., Khoshnevies, B.: Generation of alternative process plans in integrated manufacturing systems. Journal of Intelligent Manufacturing 14 (2003)
4. Sormaz, D., Ganduri, J.: Integration of Rule-based Process Selection with Virtual Machining for Distributed Manufacturing Planning. Springer, London (2007)
5. Błażewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Węglarz, J.: Handbook on Scheduling From Theory to Applications. In: Series: International Handbooks on Information Systems, Sringer, Heidelberg (2007)
6. Krenczyk, D.: Production flow synchronisation in the manufacturing assembly systems. In: Worldwide Congress on Materials and Manufacturing Engineering and Technology, COMMENT 2005, Gliwice - Wisla, Poland, p. 178 (2005)
7. Kalinowski, K.: Scheduling of production orders with assembly operations and alternatives. In: Proc. Int. Conf. Flexible Automation and Intelligent Manufacturing, FAIM 2009, University of Teesside, Middlesbrough, UK, p. 85 (2009)

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

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

Hasso-Plattner-Institut, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
{Theodor.Heinze,Robert.Wierschke,Alexander.Schacht,
Martin.vonLoewis}@hpi.uni-potsdam.de

Abstract. Advancements in the development of medical apparatuses and in the ubiquitous availability of data networks make it possible to equip more patients with telemonitoring devices. As a consequence, interpreting the collected data becomes an increasing challenge. Medical observations traditionally have been interpreted in two competing ways: using established theories in a rule-based manner, and statistically (possibly leading to new theories). In this paper, we study a hybrid approach that allows both evaluation of a fixed set of rules as well as machine learning to coexist. We reason that this hybrid approach helps to increase the level of trust that doctors have in our system, by reducing the risk of false negatives.

1 Introduction

With the changing demographics in Germany, the number of elderly people that the health care system must support increases. Furthermore, Germany faces the problem that specialists such as cardiologists move away from rural areas requiring patients from those areas to travel long distances to see them. Therefore, even a simple check up involves significant effort. In the context of the Fontane [1] project, we plan to use mobile communication technology to develop new solutions that overcome these limitations. The main idea is to equip patients that require regular observation of their vital signs, e.g. patients with heart diseases, with sensors that collect medically relevant information. Such sensors include scales and blood pressure monitors, a simple self-evaluation of a patient's wellbeing as well as more complex sensors such as electrocardiogram recorders. The measurement data is collected by a special home broker device that transmits the data to a telemedicine center (TMC). In the telemedicine center the patient's vital sign will be analyzed by medical experts. Furthermore, the data is available to other persons involved in the patient's care, e.g. the family doctor. Thus, patients can be monitored even in rural areas without the need to travel long distances or even be hospitalized. This not only saves time and money, but also improves the health care itself and provides the patient with the comfort of their own home. The Fontane monitoring scenario is depicted in Figure 1.

In the context of Fontane, we expect a ratio from 5-10 doctors for about 1000-5000 patients to be monitored during three medical studies. In order to cope with the raising number of patients that require telemedical monitoring, the measurement data needs

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 413–420, 2011.

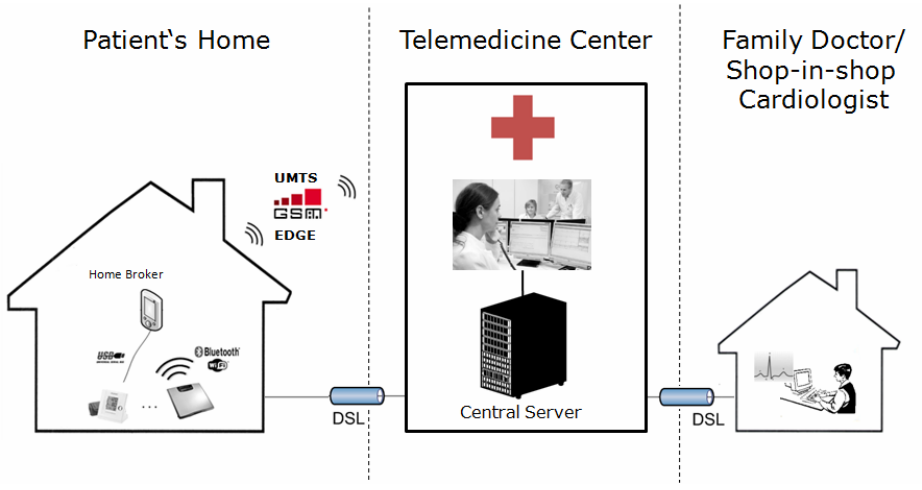


Fig. 1. Fontane Architecture

to be prioritized. We are currently developing a self-adaptive, prioritizing middleware (SaPiMa) that integrates hybrid artificial intelligence techniques [8] in order to suggest a review order of the patient data. Additionally, the middleware uses the decisions made by the doctors to improve its suggestions. Hence, more patients can be monitored while patients with critical conditions are noticed early. The main contribution of this paper is the introduction of an architecture of a hybrid artificial intelligence system for telemonitoring of heart patients.

2 Architecture

In the Fontane scenario described in Section I, the patient uses a number of different measurement devices. The types of the devices (e.g. blood pressure monitor, ECG recorder) vary depending on the therapy, which defines the measurement plan. After completion of the measurement the devices transmit the data to the home broker via Bluetooth. The home broker acts as the central hub in the patient's home. Upon receiving the measurements from the various devices, the home broker converts the device specific data formats and transmits the data to the telemedicine center using mobile networks. On the side of the telemedicine center, the incoming data will be stored in an electronic health record (EHR). In order to provide interoperability with various EHR systems and medical data standards, the measurements are processed by a J2EE-based SaPiMa Endpoint (see Figure 2). The SaPiMa Endpoint uses the Apache Camel framework [2]. Camel allows the declarative configuration of routes and processors that are used to handle messages. Due to its declarative approach the system can be easily reconfigured and extended. The SaPiMa Endpoint converts the measurement according to the required record format. Additionally, the measurements are processed by a prioritization engine. According to the calculated priority, the SaPiMa system will suggest a review order for the patients. Furthermore, SaPiMa will

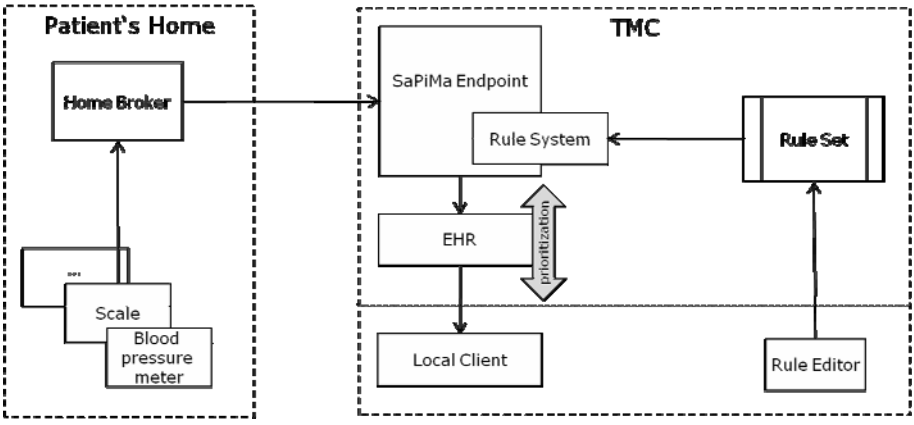


Fig. 2. SaPiMa testbed

learn from the actions of the doctors to improve the prioritization in the future. Due to the prioritization, critical changes in the health status of a patient can be noticed early which increases the survivability of the patient.

We have set up the system depicted in Figure 2 in a testbed for further research and added a neural network which, together with the rule system, builds a hybrid prioritizer. The following sections explain in detail the implementation of our hybrid artificial intelligence prioritization engine.

3 Concept of a Hybrid Artificial Intelligence Classifier

3.1 Challenges

In order to provide a prioritization as described above, SaPiMa makes high demands on the utilized classifier. Vital parameters, such as blood pressure and body weight are transmitted according to a measurement schedule. In the TMC, SaPiMa calculates a priority based on the measurement values. Thus, a review order for the patients is suggested in order to assist the medical experts. To increase the prioritization accuracy, not only current data but also historical and master data are taken into account. However, the data available for the prioritization differs depending on patient and situation. Hence, the prioritization algorithm must be flexible enough to deal with variable input vectors.

Furthermore, the prioritization mechanism must be able to interpret the different values correctly, e.g. distinguish a body weight from a pulse rate value. This can be achieved by using a classification system. In the context of the Fontane project we augment each measurement value with a LOINC code [3].

Another important aspect is the statistical distribution of the aberrant measurement values. According to oral reports of experienced telemedicine cardiologists, only about 1-5% of all patient data reviews result in an intervention. In the majority of cases, no further action is taken. This distribution poses a particular challenge for a statistical priority estimator because it obtains a >90% success ratio simply by

guessing the same every time. The training process of a classifier like a multilayer perceptron would be, without further modifications, very susceptible to terminating the training process at that point.

At the same time, this 1-5% of the patients requires a reliable monitoring due to their state of health. The goal is to eliminate the false negative error (sick patient considered as healthy), at the cost of the false positive error (healthy patient considered as sick). While a classifier with a high false negative error rate might lead to worsening of healthcare provision, a high false positive error rate makes the system less efficient with regard to working time of the telemedicine center personnel.

Another challenge is the explanation system. Due to legal and ethical reasons the ability to explain its decisions would be a great benefit for the prioritization mechanism. For a rule-based system, decision explanation would be very easy by just listing the set of matching rules, whereas statistical classifiers give results that are difficult to explain automatically [6]. On the other hand, rule-based systems are, as a general rule, not as efficient as probabilistic ones: it is tedious to construct them manually, and they grow significantly in size if derived automatically.

In summary, neither a rule-base nor a static classifier satisfies the requirements for remote patient monitoring. For our middleware, we developed a hybrid classification system which combines both approaches in order to overcome their shortcomings.

3.2 Concept of a Hybrid Classifier

Since both approaches offer numerous advantages, we propose a classification system which combines both a rule-based and a statistical classifier thus stacking some benefits and eliminating some shortcomings. The goal is to achive the precision and flexibility of a statistical classifier as well as the controllability of a rule-based one.

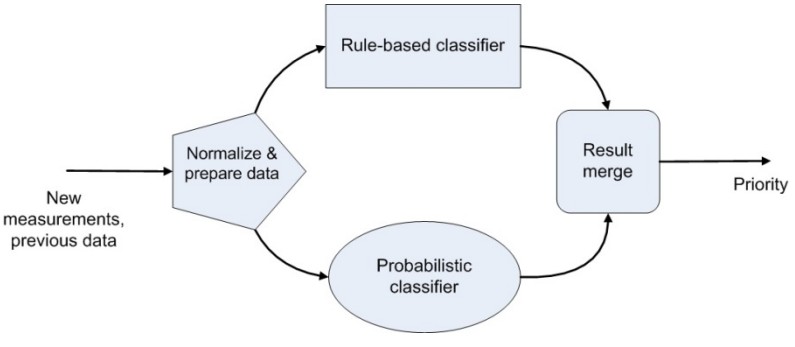


Fig. 3. Hybrid Classifier

Figure 3 shows the conceptual design of the system. The incoming data consists of current and historical measurements sorted by LOINC-Codes. The data is passed to both the rule-based and the statistical classifier. Both then determine a prioritization value. The output of the statistical classifier is a very exact estimation of the abnormality of the given measurement data. It is a real number in the range between zero (= absolutely normal) and 1 (= totally abnormal). The statistical classifier uses machine learning techniques to learn from former classification decisions met by the

doctor to learn to fully imitate his decision behavior. There is also a strong focus on avoiding false negative errors. Currently, an artificial neural network (feed-forward MLP with backpropagation) with loss matrix is used.

The output of the rule-based classifier is a Boolean value which is zero (normal) or 1 (abnormal). The rule-based classifier relies on a set of rules created and maintained by the doctors. This allows to influence the classification process and adapt the prioritization to patient-specific cases that the statistical classifier does not recognize for some reason.

The overall output of the system is computed as follows:

```
output = maximum(output_statistical, output_rule_based)
```

We hope that the statistical classifier produces reliable results in the majority of the cases, where the rule-based one deals with outliers.

4 Rule-Based Classifier

The core of a rule-based or manual classifier is a set of rules. These rules must be created and managed by domain experts (physicians or study nurses). Therefore, we developed a domain-specific language (DSL) that empowers the medical experts to specify the rule set using a familiar terminology. The language is based on predicate logic [4] and allows the specification of generic as well as patient-specific thresholds.

Additionally, we developed an editor to edit the rule set easily as possible. The process of classification starts with creating or editing the rule set with the aid of the described editor. After that, the rule set is exported into a rule interpreter. This interpreter classifies the measured values on the basis of the rule set. The language and its automatic generated editor are implemented using Xtext [5].

5 Probabilistic Classifier

As described in the challenges chapter, the prioritization may not produce false negative errors for the sake of healthcare on the one hand and should on the other hand keep the false positive error rate as low as possible to reduce the amount of time doctors spend reviewing patient measurements. To deal with asymmetries in the weighting of the errors, a loss matrix is used to adjust how the particular errors are weighed. More specifically, a doctor could adjust a loss matrix to prefer additional effort (more reviews) over false negative errors (healthcare), to prevent measurement data being classified as low priority.

In order to come up with a good classifier, proper selection of the input vector is important. Several input vectors should be considered to choose the one that leads to the best classification result. On the one hand, the optimum input vector is the one that uses all available relevant information. On the other hand, the data available may differ. Hence, the classifier should be able to deal with missing data.

As a strategy, we use a static input vector which is semantically divided into slots. A slot is a group of input neurons which receives input data from a single data entity e.g. a particular measurement and, in case of a perceptron, addresses several input

neurons. For every measurement type like weight, blood pressure etc. *n* slots are reserved. Each slot contains information such as measurement value, measurement age in hours and, very important, availability of the particular information. Slots reserved for a particular measurement type are filled with the according last *n* measurements, e.g. there are some slots reserved for blood pressure, marked by "LOINC 8480-6". The rare case of missing information is dealt with by simply marking the availability input neuron. Of course, the perceptron needs to be trained accordingly to be prepared for missing information. The structure of the input vector is depicted in Figure 4.

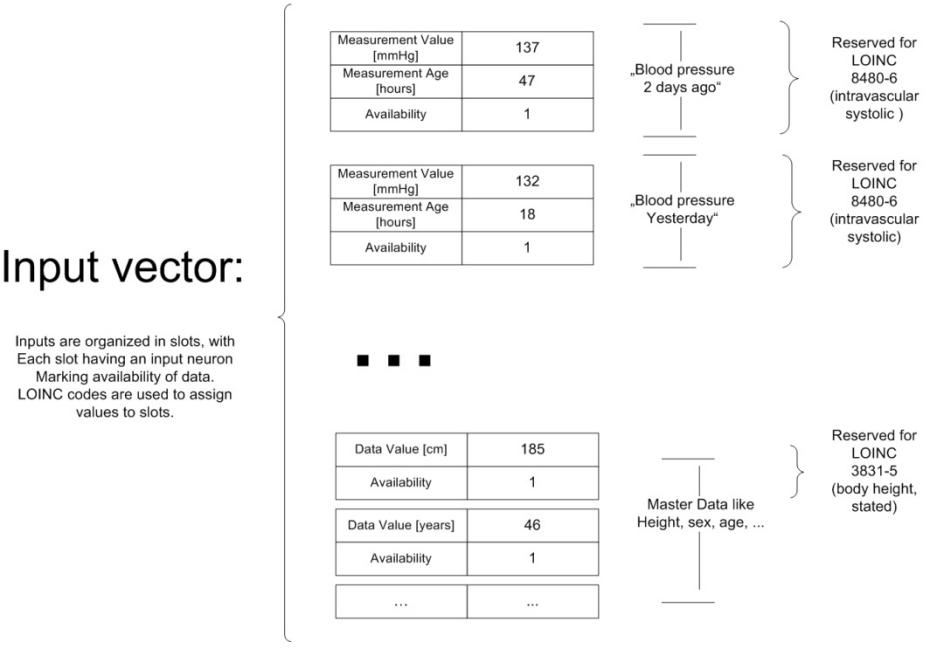


Fig. 4. Input Vector Construction

For a given data set, a suitable topology must be determined for the classifying MLP. It is necessary to perform several training runs with varying number of hidden neurons, before committing to the most promising particular topology.

6 Experimental Results

For experimenting purposes, we have generated artificial patient measurement data. We used parts of data from the "Heart Disease Data Set" (Courtesy of Cleveland Clinic Foundation) found in the UCI machine learning repository [7] and added some noise to create artificial timelines of patient monitoring data. Additionally, artificial events have been added by randomly increasing weight and decreasing blood pressure for a period of 11 days, peaking in a (noisy) 5% change in day 6. Then we have run simulations on how effectively these events are recognized by both statistic and rule-based classifiers alone and in combination.

The statistic classifier consists of a MLP with one hidden layer and a random nodes count, selected from 10 training runs. The input values are normalized to be in the range 0…1. Target (priority) values are 1 in case of an event, >0 if the day lies within the 11 days escalation period and 0 else.

The rule-based classifier consists of the following two rules that are arbitrarily derived from the artificial event generation process. Return value is 1 if a rule matches and 0 else.

```
IF last_weight > 110% of mean_weight
                              THEN recommend_hospitalization

IF last_blood_pressure < 90% of mean_blood_pressure
                              THEN recommend_hospitalization
```

Table 1. Results

	Neural network classifier	Rule-based classifier	Combined classifier
Av. Error %	**1.25**	18.70	2.05
false negative %	23.82	30.31	**17.98**

While the overall error ratio of the combined classifier is slightly worse than the error ratio of the neural network alone (Table 1), the ratio of patients prioritized too low is considerably lowered by the hybrid model.

7 Conclusion and Outlook

We have presented the core functionalities of the middleware architecture in the Fontane project: a network of medical telemonitoring devices, and a telemedicine center in which the measurements are classified with the objective of determining patients for whom a medical intervention is necessary. As the number of patients increases, inspecting daily measurement results become monotonous and requires more personnel. Using machine-based classification can help to reduce the risk of mistakes as well as to make doctors more productive, giving them more time for the relevant cases.

Our system is designed to be independent from and adaptable to the specific decease and therapy. To achieve this objective, both data models and classification machinery must be flexible. We have presented approaches to achieve this flexibility by supporting open data models, in-field updates to “static” rules, and machine learning. Currently, we consider the integration of a case based reasoning component as described in [9].

As an evaluation, we have presented results of feeding test data into our implementation. In the coming years, we will employ this implementation in field studies involving several hundred patients; this should give more insights into the practical operation of our hybrid approach to artificial intelligence.

References

1. Polze, A., Tröger, P., Hentschel, U., Heinze, T.: A Scalable, Self-Adaptive Architecture for Remote Patient Monitoring. In: 2010 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, isorcw, pp. 204–210 (2010)
2. Apache Camel Framework, http://camel.apache.org/
3. McDonald, C.J., Hu, S.M., Suico, J.G., Hill, G., Leavelle, D., Aller, R., Forrey, A., Mercer, K., DeMoor, G., Hook, J., Williams, W., Case, J., Maloney, P.: LOINC, a universal standard for identifying laboratory observations:a 5-year update. Clinical Chemistry 49(4), 624–633 (2003), http://www.ncbi.nlm.nih.gov/pubmed/12651816, PMID: 12651816
4. Levy, A.Y., Rousset, M.C.: CARIN: A representation language combining horn rules and description logics. In: ECAI, vol. 1996, pp. 323–327 (1996)
5. Xtext Language Development Framework, http://www.eclipse.org/Xtext/
6. Bishop, C.: Pattern Recognition and Machine Learning. Springer, New York (2006)
7. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/
8. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
9. Ashwin Kumar, K., Singh, Y., Sanyal, S.: Hybrid approach using case-based reasoning for domain independent clinical decision support in ICU. Expert Systems with Applicaton 36, 65–71 (2009)

Hybrid Patient Classification System in Nursing Logistics Activities

Dragan Simić[1], Dragana Milutinović[2], Svetlana Simić[2,3], and Vesna Suknjaja[3]

[1] University of Novi Sad, Faculty of Technical Sciences,
Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia
dsimic@eunet.rs
[2] University of Novi Sad, Faculty of Medicine,
Hajduk Veljkova 3, 21000 Novi Sad, Serbia
milutind@uns.ac.rs
[3] Clinical Centre of Vojvodina, Clinic for Neurology,
Hajduk Veljkova 1-9, 21000 Novi Sad, Serbia

Abstract. The history of patient classification in nursing dates back to the period of Florence Nightingale. The first and the foremost condition for providing quality nursing care, which is measured by care standards, and determined by number of hours of actual care, is the appropriate number of nurses. Patient classification criteria are discussed in this paper. Hybrid classification model based on learning vector quantization (LVQ) networks and self-organising maps (SOM) are purposed. It is possible to discus three types of experimental results. First result could be assessment of Braden scale and Mors scale by LVQ. Second result, the time for nursing logistics activities. The third is possibility to predict appropriate number of nurses for providing quality nursing care. This research was conducted on patients from Institute of Neurology, Clinical Centre of Vojvodina.

Keywords: Patient classifications, logistics activities, nursing, hybrid system.

1 Introduction

Questions concerning patient safety and the quality of nursing care have been raised by quick and dynamic changes in the health care domain [1]. The first and the foremost condition for providing quality nursing care, which is measured by care standards, and determined by number of hours of actual care, is the appropriate number of nurses [2]. The continuous challenge here is to establish an objective and reliable way for determining how many nurses will be needed to satisfy patients' needs, wishes and expectations. With this purpose, many different classification systems (grouping of patients into a certain category for a certain purpose) that is, classification of patients according to their need for care, have been created. Patient classification provides a quick insight into the gravity of condition of a ward patient, and indicates what kind of care is to be given to that patient, and how many nurses will be necessary for adequate health care.

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 421–428, 2011.

The rest of the paper is organised as follows: the following sections provide a brief overview of hybrid artificial intelligence systems and some background patient classification system in nursing logistics activities. The section 3 overviews challenges in modelling patient classification, while section 4 elaborates concept and designs of patient classification in Republic of Serbia and Republic of Croatia. Section 5 shows purpose hybrid classification model, implemented methods and usage data set. The section 6 describes experimental results while section 7 overviews future work and concludes the paper.

2 Background

Artificial intelligence techniques have demonstrated a capability to solve real-world problems in science, business, technology, and commerce. The integration of different learning techniques and their adaptation, which overcomes individual constraints and achieves synergetic effects through hybridisation or fusion, has in recent years contributed to a large number of new intelligent system designs [3].

The hybridisation of intelligent techniques, drawn from different areas of computational intelligence, has become prevalent because of the growing awareness that they outperform individual computational intelligence techniques. In a hybrid intelligence system, a synergetic combination of multiple techniques is used to build an efficient solution to deal with a particular problem [4]. Evolutionary algorithms, instance selection and feature selection as the most known techniques, for data reduction in data mining problems have been successfully used. Their aim is to eliminate irrelevant and/or redundant features and to obtain a simpler classification system. This reduction can improve the accuracy of this model in classification [5].

There are some studies where patient classification systems are discussed. Development of a prototype patient classification instrument designed specifically for rehabilitation patients is focus in [6]. This research is based on and continues one of the early paper about *Nursing workload measurement as management information* [7]. The process of instrument development strategies is discussed and it includes: staff education, management support, data analysis including the development of supporting information systems, and ongoing use of the rehabilitation patient classification system.

Personal dimensioning in psychiatric nursing motivated the development of an instrument to classify the level of dependence in psychiatric nursing based on statistical methods, Kappa coefficient and the Spearman correlation is proposed in [8]. The emergency department is a dynamic environment with a high throughput of patients. In order to provide optimal care for patients a responsive staffing pattern is required, according to the clinical stability of patients which can vary considerably. In [9] twelve patients classification systems are discussed, but only three systems reported evidence of good validity and reliability: the *ED Patient Needs Matrix* developed in the US, the Conner's Tool (a modified version of the *Patient Needs Matrix*) developed in Australia and the Jones Dependency Tool developed in UK.

3 Challenges in Modelling Patients Classification

The history of patient classification in nursing dates back to the period of Florence Nightingale, when an informal classification method reflecting nursing workload was used. Based on intuition, perhaps, the most seriously ill patients on the large open Nightingale wards were placed near to the ward sister's office to facilitate their observation. On the other hand, those patients who could take care of themselves tended to be located at the far end of the ward, indicating their decreased dependency on the nursing staff [10]. During 1950s and 1960s great emphasis and attention have been given to adequate health care because of greater health care costs and the shortage of working force [11].

Table 1. Patient classification by Mary Ellen Warstler

	Category	Hours	Average hours
I	Self care	1 to 2	1.5
II	Minimum care	3 to 4	3.5
III	Intermediate care	5 to 6	5.5
IV	Modified intensive care	7 to 8	7.5
V	Intensive care	10 to 14	12

In 1973 Mary Ellen Warstler defined 5 patient categories according to the care needed in 24 hours which is presented in Table 1 [12].

4 Concept and Designs of Patient Classification

It is important to determine classification criteria in order to place patients in categories according to the amount of health care that is necessary. These criteria indicate the states or activities that most influence time spent on providing adequate health care [10]. The choice of criteria for patient classification is based on Virginia Henderson's health care definition, which defines the role of a nurse as providing assistance to a patient in satisfying 14 basic human needs, and Dorothea Orem's health care definition which is based on the idea of self – care [13].

When choosing classification criteria concerning patient health care needs, therapeutic and diagnostic procedures should be taken into account. When classifying patients, authors use two approaches – description of specific criteria characteristic to certain category - an example of patient classification in Republic of Serbia. The separate scoring of individual criteria and calculating average value which then represents patient category - an example of patient classification in Republic of Croatia.

Patient classification criteria in Republic of Serbia, with the purpose of providing quality health care, are placed in five categories: 1) *General care*; 2) *Semi-intensive care*; 3) *Intensive care*; 4) *Special intensive care*; 5) *Special care* [14].

General care is defined with the following classification criteria: (1) Preserved consciousness; (2) Time - space orientation; (3) Vital signs are checked every 12 hours; (4) Absence of bleeding; (5) Ability to move; (6) Ability to feed themselves; (7) Does not regurgitate; (8) Able to satisfy physiological needs without assistance;

(9) Able to maintain personal hygiene; (10) Does not need other person's help; (11) Adequate medical treatment. Six of these criteria are sufficient to define patients in g*eneral care*.

Other four patient categories are based on the same type of definition, for each category respectively. Peculiarity of this classification is that the decision on the degree of necessary health care is based on the fulfilment of a number of linguistics characteristics.

Patient classification criteria in Republic of Croatia. Patients are placed in five categories under this classification there are: 1) *Self care*; 2) *Minimum care*; 3) *Intermediate care*; 4) *Intensive care*; 5) *Special care* [15]. For all patient categories 16 patient's activities are defined and rated on 4-point scale.

Evaluation of patient's ability to maintain his/her: (1) Personal hygiene; (2) Dress; (3) Feed; (4) Evaluation of elimination – other person's assistance; (5) Walking and standing; (6), Sitting; (7) Moving and Turning; (8) Risk of falling; (9) State of consciousness; (10) Pressure sores risk; (11) Vital signs; (12) Communication; (13) Specific health care procedures; (14) Diagnostic procedures; (15) Therapeutic procedures; (16) Education undertaken and type of knowledge.

Risk of falling - If there is no risk of falling, the patient is placed in Category 1, if there is a risk of falling then the patient is evaluated by *Mors* scale. Depending on the number of points, patients can be placed in three following categories (sub-selections): low risk (0-24 points), moderate risk (25-44 points), and high risk (44 plus points). Successfully conducted research on risk of falling of clinical patients is presented in [16].

Pressure sores risk - is evaluated by *Braden* scale. Depending on the number of points on the scale, patients are placed in five sub-selections (categories). According to *Braden* scale there are following categories: no risk (19-23 points), risk present (15-18 points), moderate risk (13-14 points), high risk (10-12 points) and very high risk (9 points and less).

According to *Critical factors table* in patient classification, *Braden* scale and *Mors* scale included, each of 16 factors of classification can be evaluated on 4-point scale. This means that overall minimum number of points can be 16 and overall maximum number of points can be 64. Modified normal distribution was used when placing points into categories. Finally, point distribution for each category is following: *First* category 16 – 26 points; *Second* category 27 – 40; *Third* category 41 – 53; *Fourth* category 54 – 64 points.

5 Hybrid Classification Model, Methods and Material

The aim of this research is to evaluate patient categories and amount of health care, then determine the number of hours of actual care, and at the end, the appropriate number of nurses for providing quality nursing care. We purpose hybrid artificial neural networks (ANN) for this complex analysis. As with any system, simple ANN has its limitations: (1) The learning stage can be very drawn out; (2) The system might not achieve a stable absolute minimum configuration, but could stay with local minimums; (3) The system may begin to oscillate in the learning phase; (4) It is necessary to repeat the learning phase when significant changes take place in the actual situation; (5) The analysis of the weightings is complex and difficult to interpret.

Two of the most used competitive ANN algorithms used are self-organizing maps (SOM) and learning vector quantization models (LVQ). SOM and LVQ models have been successfully used in different scientific fields [17]. According to previous limitations in this research the great advantages of a hybrid ANN model is proposed, particularly LVQ and SOM.

Learning vector quantization networks can classify any set of input vectors, not only linearly separable sets. The only requirement is that the competitive layer must have enough neurons, and each class must be assigned enough competitive neurons. LVQ models classify input vectors into target classes by using a competitive layer to find subclasses of input vectors, and then, combine them into the target classes defined by the user.

Self-Organising Map algorithm is probably the best known ANN technique. It is based on type of unsupervised learning called competitive learning, and adaptive process in which the neurons in neural networks gradually become sensitive to different input categories or sets of samples in a specific domain of the input space.

Data set - Patient classification criteria characteristic for Republic of Croatia is discussed in this research. This patient classification is better for hybrid ANN model then patient classification criteria in Republic of Serbia. The data set is used from Institute of Neurology, Clinical Centre of Vojvodina, Novi Sad, Serbia. The data set is collection of data of 27 different patients who have been observed for two weeks, these data then determined classification criteria in order to place patients in categories according to the amount of health care that is necessary.

Education undertaken and type of knowledge now become the input to the SOM. Patients are placed in five categories under this classification: 1) *Self care*; 2) *Minimum care*; 3) *Intermediate care*; 4) *Intensive care*; 5) *Special care*. Then hybrid LVQ-SOM is used to predict appropriate number of nurses for providing quality nursing care.

6 Experimental Results

Risk of falling and *Pressure sores risk* are classified by LVQ networks. The output of *Risk of falling* and *Pressure sores risk* data that are used in LQW now become the input to the SOM with other 14 input data: 1) *Personal hygiene*; (2) *Dress*; (3) *Feed*; (4) *Evaluation of elimination*; (5) *Walking and standing*; (6), *Sitting*; (7) *Moving and Turning*; (9) *State of consciousness*; (11) *Vital signs*; (12) *Communication*; (13) *Specific health care procedures*; (14) *Diagnostic procedures*; (15) *Therapeutic procedures*; (16) *Education undertaken and type of knowledge* now become the input to the SOM. Then hybrid LVQ-SOM is used to provide the time needed for nursing logistics activities and to predict appropriate number of nurses for providing quality nursing care.

The basic usage of the SOM has following steps: (1) construct data set; (2) normalise it; (3) train the map; (4) visualise map; (5) analyse results. The batch training algorithm is used in this process [18]. SOM is used for probability density estimation. Each map prototype is the centre of a Gaussian kernel whose parameters are estimated from the data. The Gaussian mixture model is estimated and the probabilities can be calculated. The map grid is in the output space.

Table 2. A part of input data set – for one week

		Mo	Tu	We	Th	Fr	Sa	Su
General care	patients	9	10	10	10	10	10	10
(1 hour)	hours	9	10	10	10	10	10	10
Semi-intensive care	patients	8	7	7	7	8	8	8
(3 hours)	hours	24	21	21	21	24	24	24
Intensive care	patients	3	3	3	3	2	2	2
(6 hours)	hours	18	18	18	18	12	12	12
Special intensive care	patients	0	0	0	0	0	0	0
(12 hours)	hours	0	0	0	0	0	0	0
Special care	patients	0	0	0	0	0	0	0
(24 hours)	hours	0	0	0	0	0	0	0
Total number of patients		20	20	20	20	20	20	20

Table 3. Calculated and LVQ-SOM results – for nursing logistics activities – for one week

	Mo	Tue	We	Thu	Fr	Sa	Su
Total number of patients	20	20	20	20	20	20	20
Number of hours of necessary care	51	49	49	49	46	46	46
Number of nurses in 3 shifts	5	5	5	5	5	4	4
Number of hours (nurses x 8 hours)	40	40	40	40	40	32	32
Variation (+ / -)	-11	-9	-9	-9	-9	-12	-12
Number of nurses – minimum	6.4	6.1	6.1	6.1	5.8	5.8	5.8
Number of nurses – minimum (full-time + part time)	6.5	6	6	6	6	6	6
Number of nurses – minimum (full-time)	6	6	6	6	6	6	6
Number of nurses – average	8.6	8.2	8.2	8.2	7.8	7.8	7.8
Number of nurses – average (full-time + part time)	8.5	8	8	8	8	8	8
Number of nurses – average (full-time)	9	8	8	8	8	8	8
LVQ – SOM; assess nurses number (full time + part time)	*8*	*7.5*	*7.5*	*7.5*	*7*	*7*	*7*

It is possible to discus three types of results. First result could be assessment on *Braden* scale and *Mors* scale. This result shows that LVQ networks provide correct estimation for 95% of the cases. Second result could be assessment for providing quality nursing care, and the time for nursing logistics activities. These results could be discussed in several different ways. The basic reason is that the right time is not defined exactly, rather expected range of time (Table 3). This result can vary to a great extent and its limitations must be taken into account, and, it is very error sensitive. That is why the results in Table 3 are given for average and minimum care time. And finally, the third result predicts the appropriate number of nurses for providing quality nursing care. These estimation results depend on previous (second) results for assessment of care time, but the results gained are less error sensitive. The hybrid LQV-SOM gives experimental results which are between minimum and average

hours for nursing logistics activities. The main reason is that the predicted appropriate number of nurses for providing quality nursing logistics care must be an integer.

Moreover, this research shows how necessary number of nurses for providing quality nursing care differs from the actual number of hired nurses. This represents a very important conclusion of our research. It is a well known fact that workers do not have to be hired full time, but that they could be hired only part time. This means that, after quality nursing care assessment has been conducted, there is no need to hire a full time nurse, and that it is possible to optimize care time and expenses only for necessary logistic activities. To improve presented experimental results one innovative research area that uses fusers as part of a multiple classifier system, which is discussed in Designing fusers on the basis on discriminants – evolutionary and neural network methods [19], could form part of future research in hybrid patient classification system.

In general, tools and instruments for the type of classification and patient follow up, although well described and useful in clinical practice, are not represented in a satisfying way. Awakening awareness about its usefulness through interdisciplinary studies and medical stuff, physicians and nurses [20], collaboration can make significant contribution to wide spread acceptance of this method. Contribution of this paper presents a way to use artificial intelligent system, hybrid purpose ANN model, particularly LVQ-SOM instead of statistical methods that have been widely used so far to solve real-world problem of nurses.

7 Conclusion and Future Work

Hybrid classification model based on LVQ-SOM networks is proposed in this paper. The foremost condition for providing quality nursing care is the appropriate number of nurses. Assessment for *Braden* scale and *Mors* scale uses LVQ networks. The first result provides correct estimation for 95% of the cases. Second result provides the time needed for nursing logistics activities. The third result is able to predict appropriate number of nurses for providing quality nursing care. Experimental results are shown not just in respect to minimum hours for nursing logistics activities, but for the average as well. The acquired experimental results used by hybrid classification model LVQ-SOM are between minimum and average values for patient classification criteria in Republic of Croatia. This research was conducted on patients from Institute of Neurology, Clinical Centre of Vojvodina.

Although the experimental results we have gained are valid, the research of hybrid classification systems could be continued. Presented model for classification is not limited to this case study. Also, this research could be improved by soft computing techniques utilization, as well as fuzzy logic and genetic algorithm utilization.

References

1. Rimac, B., Bišćan, J.: Nursing Paradigm and Patient Safety. Medix 86, 167–170 (2010)
2. Twig, D., Duffield, C.: A Review of Workload Measures: A Context for a New Staffing Methodology in Western Australia. International Journal of Nursing Studies 46, 132–140 (2009)

3. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid Learning Machines. Neurocomputing 72(13-15), 2729–2730 (2009)
4. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid Intelligent Algorithms and Applications. Information Science 180(14), 2633–2634 (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(LNAI), vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
6. Sarnecki, A., Haas, S., Stevens, K.A., Willemsen, J.: Design and Implementation of a Patient Clasification System for Rehabilitation Nursing. Journal of Nursing Administration 28(3), 35–43 (1998)
7. Vries, G.: Nursing Workload Measurement as management Information. European Journal of Operational Research 29, 199–208 (1987)
8. Martins, P.A., Arantes, E.C., Forcella, H.T.: Patient Classification System in Psychiatric Nursing: Clinical Validation. Journal of School Nursing 42(2), 223–241 (2008)
9. Williams, S., Crouch, R.: Emergency Department Patient Classification System: A Systematic Review. Accident and Emergency Nursing 14(3), 160–170 (2006)
10. Giovannetti, P.: Understanding patient classification systems. Journal of Nursing Administration 9(2), 4–9 (1979)
11. Ćorluka, V.: Standardized care purport – new nursing care quality. Health Care 34(3), 25–35 (2005)
12. Warstler, M.E.: Cyclic work schedules and a nonnurse coordinator of staffing. Journal of Nursing Administration 3(6), 45–51 (1973)
13. Wills, E.M.: Grand Nursing Theories Based on Human Needs. In: McEwen, M., Wills, E. (eds.) Theoretical Based for Nursing. Lippincot Williams & Wilkins (2007)
14. Ćorluka, V., Aleksić, Ž., Savićević, M.: Instruction on Record Keeping in Health Protection, Family Care, Field Nurse Care and Health Care. Federal Institute for Health Improvement and Protection, Beograd (2000)
15. Croatian Chamber of Medical Nurses. Patient Classification According to Their Health Care Needs (2006), http://www.hkms.hr
16. Milutinović, D., Martinov-Cvejin, M., Simić, S.: Padovi i povrede hospitalizovanih pacijenata kao pokazatelji kvaliteta rada bolnice. Medicinski pregled LXII (5-6), 249–257 (2009)
17. Torrecilla, J., Rojo, E., Oliet, M., Domínguez, J.C., Rodrínguez, F.: Self-Organizing Maps and Learning Vector Quantization Networks as Tools to Identify Vegetable Oils and Detect adulterations of Extra Virgin Olive Oil. In: Pierucci, S. (ed.) ESCAPE, vol. 20, Elsevier, Amsterdam (2010)
18. Simić, D., Kovačević, I., Simić, S.: Insolvency Prediction for Assessing Corporate Financial Health. Logic Journal of IGPL (2011), doi:10.1093/jigpal/jzr009
19. 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)
20. Simin, D., Milutinović, D., Brestovački, B., Simić, S., Cigić, T.: Attitude of health science students towards interprofesional education. HealthMED Journal 4(2), 461–469 (2010)

An Approach of Soft Computing Applications in Clinical Neurology

Dragan Simić[1], Svetlana Simić[2], and Ilija Tanackov[1]

[1] University of Novi Sad, Faculty of Technical Sciences,
Trg Dositeja Obradovića 6,21000 Novi Sad, Serbia
dsimic@eunet.rs, ilijat@uns.ac.rs
[2] University of Novi Sad, Faculty of Medicine,
Hajduk Veljkova 1, 21000 Novi Sad, Serbia
drdragansimic@gmail.com

Abstract. This paper briefly introduces various soft computing techniques and presents miscellaneous applications in clinical neurology domain. The aim is to present the large possibilities of applying soft computing to neurology related problems. Recently published data about use of soft computing in neurology are observed from the literature, surveyed and reviewed. This study detects which methodology or methodologies of soft computing are frequently used together to solve the specific problems of medicine. Recent developments in medicine show that diagnostic expert systems can help physicians make a definitive diagnosis. Automated diagnostic systems are important applications of pattern recognition, aiming at assisting physicians in making diagnostics decisions. Soft computing models have been researched and implemented in neurology for a very long time. This paper presents applications of soft computing models of the cutting edge researches in neurology domain.

Keywords: Soft Computing, neurology, neural networks, fuzzy logic.

1 Introduction

In recent years more sophisticated methods that can model non-linear, complicated, real-world applications are needed. Soft computing (SC) methods have been successfully applied to solve non-linear problems in engineering, business and medicine. These methods, which indicate a number of methodologies used to find approximate solutions for real-world problems which contain kinds of inaccuracies and uncertainties, can be alternative to statistical methods. The underlying paradigms of soft computing are neural computing, fuzzy logic computing and evolutionary computing [1].

Neurology is a medical speciality dealing with nervous system disorders. More specifically, it deals with diagnosis and treatment of all categories of disease involving the central, peripheral, and autonomic nervous systems, including their all effectors tissue such as muscle.

The paper surveys the usage of SC in neurology domain which, in general, includes: fuzzy logic (FL), artificial neural networks (ANNs), and evolutionary computing (EC).

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 429–436, 2011.

This is also one of our purposes in this paper: to present some SC methodologies available, to represent and manage imperfect knowledge in neurology domain, where a problem can be solved more effectively by using FL, ANN and EC in combination than using any of these individually.

The rest of the paper is organised as follows. The following section presents early statistical based researches in neurology as medical discipline and the advances of artificial intelligence to solve real-world problems, while section 3 elaborates basic approach of soft computing, fuzzy logic, artificial neural networks, and evolutionary computation in the neurology domain. Section 4 shows some successful hybrid implementation of soft computing techniques in neurology implementation. Section 5 describes some future work for development and implementation of soft computing and hybrid models in neurology domain.

2 Statistics vs. Hybrid Artificial Intelligent Systems

In general, the earliest researches in medical domain used statistical methods. It can be considered that the first medical-statistical book was laid in the 17th century with publication of *Natural and Political Observation upon the Bills of Mortality* by John Graunt. Nowadays, large number of well-known statistical tests and procedures in medical domain are used as well as: multivariate analysis of variance as generalised form of univariate analysis of variance, chi-square test, correlation, factor analysis, mean square weighted deviation, Pearson correlation coefficient, regression analysis, Spearman's rank correlation coefficient. There are large numbers of neurology research papers which used statistics in different way. It can be mentioned that only some of them such as factor analysis are used in: (1) extracting salient features from EEG data as spatial information in a few factors and reducing redundancy of multichannels computerised EEG signal [2]; (2) evaluation of the Dutch version of the Parkinson's disease questionnaire 39 (PDQ39-DV) [3]; CT angiography in acute stroke to provide additional information on occurrence of infarction and functional outcome after 3 months [4]. Also, Pearson correlation coefficient is used in: (1) comparison of 36-item Short-Form Health Survey (SF-36) and World Health Organisation Quality of Life Assessment (100-items version) (WHOQOL-100) in patients with stroke [5]; normal-pressure hydrocephalus: white matter lesions correlate negatively with gait improvement after lumbar puncture [6]; correlation of vibratory quantitative sensory testing and nerve conduction studies in patients with diabetic peripheral neuropathy [7]. There is also a large number of papers where various statistics methods in medical domain are used.

On the other hand, artificial intelligence (AI) techniques have demonstrated a capability to solve real-world problems in science, business, technology, commerce, and medicine. The advantage of AI techniques is a completely different way of solving real-world problems in completely different manner from statistics. While statistics tries to assert the research results, AI techniques show their proactive role in medical research. AI techniques not only assert the state of facts but try to improve communication and reasoning in decision making concerning diagnosis and/or treatment. The

integration of different learning techniques and their adaptation, which overcomes individual constraints and achieves synergetic effects through hybridisation or fusion, has in recent years contributed to a large number of new intelligent system designs [8].

The hybridisation of intelligent techniques, drawn from different areas of computational intelligence, has become prevalent because of the growing awareness that they outperform individual computational intelligence techniques. In a hybrid intelligence system, a synergetic combination of multiple techniques is used to build an efficient solution to deal with a particular problem [9]. One innovative research area that uses fusers as part of a multiple classifier system, which is discussed in Designing fusers on the basis on discriminants – evolutionary and neural network methods [10], could form part of future research in hybrid neurology patient classification systems. Evolutionary algorithms, instance selection and feature selection as the best known techniques for data reduction in data mining problems have been successfully used. Their aim is to eliminate irrelevant and/or redundant features and to obtain a simpler classification system. This reduction can improve the accuracy of this model in classification [11].

3 Soft Computing

The applications of SC have proved to have two main advantages. First, it made possible solving nonlinear problems, in which mathematical models are not available. Secondly, it introduced the human knowledge such as cognition, recognition, understanding, learning, and other skills into fields of computing. This resulted in the possibility of constructing intelligent systems such as autonomous self-tuning systems, and automated design systems.

Fuzzy Logic provides a mathematical framework to treat and represent uncertainty in the perception of vagueness, imprecision, partial truth, and lack of information. Fuzzy systems are very useful not only in situations involving highly complex systems but also in situations where an approximate solution is warranted. To deal with qualitative, inexact, uncertain and complicated processes, the fuzzy logic system can be well-adopted since it exhibits a human-like thinking process [12]. FL is different from probability theory because FL is deterministic rather than probabilistic. Imprecision is modelled via fuzzy sets, linguistic variables, membership functions, inferences and de-fuzzification. These concepts are all handled in an entirely deterministic manner. Uncertainty in the membership functions in fuzzy set theory, i.e. uncertainty about the actual value of a membership function, has been addressed by type-2 fuzzy sets [13].

Artificial Neural Networks (ANNs) are a modelling methodology whose inspiration arises from a simplified model of the working of the human brain. ANNs can be used to construct models for the time series prediction, pattern classification, clustering, non-linear control, function approximation. ANNs are desirable because (1) nonlinearity allows better fit to the data; (2) noise-insensitivity provides accurate prediction in the presence of uncertain data and measurement errors; (3) high parallelism implies fast processing and hardware failure-tolerance; (4) learning and adaptability allow the system to modify its internal structure in response to changing environment; (5) generalisation enables applications of model to unlearned data.

Evolutionary Computation draws inspiration from the processes of biological evolution to breed solution to problems. Evolutionary algorithms search for the optimal solution by the number of modification of strings of bits called chromosomes. The chromosomes are the encoded form of the parameters of the given problem. In successful iterations, the chromosome corresponds to the maximum of the fitness function. Each generation consists of three phases: reproduction, crossover, and mutation. Genetic programming (GP) is an evolutionary algorithm in which the chromosome representation is based on a parse tree structure. The nodes are functions which act on the values presented at the inputs and provide the answer to the output. A population of individual genetic programs is subject to an evolutionary strategy employing selection, mutation and recombination (crossover) schemes. The combined application of variation and selection generally leads to improving fitness values in population.

4 Applications of Soft Computing in Neurology

Recent developments in medicine show that diagnostic expert systems can help physicians make a definitive diagnosis. Automated diagnostic systems are important applications of pattern recognition, aiming at assisting physicians in making diagnostics decisions. Automated diagnostic systems have been applied to a variety of medical procedures, such as electromyogram (EMG), electroencephalography (EEG).

In this section we shall describe the novel applications of SC models in neurology domain. SC models have been researched and implemented in neurology for a very long time, but we decided to present some of the cutting edge researches.

Most of these researches and implementations were conducted and published in recent years. In the following subsections they will be presented in different logistic branches such as: hybrid model based on synergy of principal component analysis, multilayer perceptrons, and support vector machine for classification of EMG signals are discussed in (1) first subsection [14].

Approximately 1% of the world population or 60 million individuals are diagnosed with epilepsy. Epilepsy is a disorder of cortical excitability and still represents an important medical problem. Therefore, the hybrid systems in classification of primary generalised epilepsy are discussed in (2) the second subsection [15] [16].

There are also some other researches which discuss different possibilities of using SC techniques for diagnosing different neurology disorders. We can mention some of the latest: (1) Recurrent neural networks for diagnosis of Carpal tunnel syndrome [17]; (2) An immune networks inspired evolutionary algorithm for diagnosis of Parkinson's disease [18]; (3) Multi-parametric classification of Alzheimer's disease and mild cognitive impairment for impact of quantitative magnetization transfer magnetic resonance imaging usage by fuzzy-c means classification algorithm [19]; (4) Rule-based fuzzy logic systems for assessment primary headaches [20], and migraine [21].

4.1 The Hybrid Model for Classification of EMG Signals

EMG is a signal obtained by measuring the electrical activity in a muscle, has been widely used both in clinical practice and in the rehabilitation field. Clinical analysis of the EMG is a powerful tool to assist the diagnosis of neuromuscular disorder. An

automatic diagnostic tool for neuromuscular disease, based on the feature extraction, classification, disease identification and muscle activity of myoelectric patterns, has been used in clinical practice.

In [14] the First Fourier transformation (FFT) is most common method for determining the frequency spectrum of the EMG signal. The frequency spectrum of EMG is used to detect muscle fatigue force production, and muscle fibre conduction velocity.

Principal component analysis (PCA) decomposes the covariance structure of the dependent variables into orthogonal components by calculating the eigenvalues and eigenvectors of the data covariance matrix. Eigenvalues assist in making decisions about number of orthogonal components that will be used in future analyses, while eigenvectors assist in determining the relationship between the original variables and these new components. Eigenvalues and eigenvectors transform the original variables space into a 'new' set of variables, called principal components (PCs).

Multilayer perceptrons (MLPs) are universal in the sense that they can approximate any continuous nonlinear function arbitrarily well on a compact interval. MLPs are used to solve a nonlinear optimisation problem, which has many local minima. Support vector machine (SVM) is recently developed pattern classification algorithm with nonlinear formulation. It is based on the idea that affords dot-products to be computed efficiently in higher-dimensional feature spaces. The classes which are not linearly separable in the original parametric spaces can be linearly separated in higher-dimensional feature space. Because of this, SVM has advantage of handling the classes with complex nonlinear decision boundaries. Moreover, there are rather few applications in bioengineering field and, in particular, in neurology [14].

In order to make meaning of EMG signal, spectral analysis should be applied to EMG signal, which is non-stationary since the algorithm of FFT is not complex. The FFT analysis produces a large number of coefficients, more rarely than the number of points in the original waveform. Therefore, the amount of FFT coefficients has to be reduced using PCA. In order to reduce the volume and to maximise information content of the input data, a compression technique has been applied prior to input of the data to the MLP and SVM classifiers. This provides data compression for both training and testing datasets, and reduces the dimensionality of the classifier network weight space as well. Training and testing on the basis of the EMG signals could contain many redundancies and nonessential information, which may lead to a lengthy training process. The concept of receiver operating characteristic (ROC) analysis was used for comparison of the diagnostic accuracy of the different classification methods and groups. Desired output values obtained from input vectors of EMG signals were determined as (1 0 0) – myopathy; (0 1 0) – neuropathy and (0 0 1) – normal vectors. That means that output layer contains 3 neurons.

In Table 1. performance values of different MLP topologies have been listed, *TrError* and *TeError* are the error values from training and test sets, respectively.

The optimum MLP architecture is 3-6-6-3 topology. The test performance of both classification systems is executed, the highest percentage of correct classification (85.42 %) is found in SVM. However, 84.16 % of classification with MLP is very close. ROC analysis results of both classifications have indicated that SVM presents a higher performance.

Table 1. Performance values in different MLP topology

Inputs	Hidden	Hidden (2)	TrError	TeError
1	1	15	0.369	0.373
3	4	5	0.243	0.237
4	12	10	0.238	0.245
3	5	4	0.237	0.243
3	7	6	0.247	0.238
3	4	3	0.249	0.240
3	3	3	0.252	0.236
3	6	6	0.241	0.241

4.2 The Hybrid Systems in Classification of Primary Generalised Epilepsy

Epilepsy is classified as either generalised or partial with several subcategories in each class. In the management of patients with established epilepsy, the concept of epilepsy syndrome based on age at onset, seizure type or types, EEG findings and etiology has been an important advancement. The correct diagnosis of a patient's epilepsy syndrome clarifies the choice of drug treatment and also allows accurate assessments of prognosis in many cases.

In both researches [15] [16] multilayer perceptron neural networks (MLPNN), which contain two or more layers, are used. Each layer consists of units which receive their input from a layer directly below and send their output to units in a layer directly above the unit. There are many training algorithms used to train an MLPNN. A significant improvement on realisation performance, in relation to well known back-propagation training algorithm, can be observed by using approaches such as namely Newton's method, conjugate gradient's, or the Levenberger-Marquardt optimisation technique which is used in this study.

The input of the MLPNNs had ten nodes representing parameters which are the age of seizure onset groups, sex, the activity properties of EEG findings, the physiological conditions of the patients during EEG, the existence of rhythmycity of the abdominal activities, the localisation of abdominal signals, hemispheric lateralisation, the frequency of abnormal waves, the duration of the abnormal signals and the loss of consciousness in the course of seizure time. The duration of the abnormal signals and the frequency of abnormal waves that are used as input to MLPNNs were interval variables.

The learning of the network was executed by applying the input and output vectors. In this classification, the output layer of MLPNN represented the subgroups of primary generalised epilepsy: (generalised tonik-klonik, absans, myoclonic and more than one type seizure). In the hidden layer and the output layer, the activation functions were selected as sigmoid and softmax function, respectively. The MLPNNs were developed using 34 training examples, while the remaining 45 examples were used for testing model. For obtaining a better generalisation, nine training examples were selected randomly to be used as a cross validation set. When the structure of the neural network was formed as a result of the performed experiments, the MLPNN having 45 nodes in the hidden layer had the best total classification accuracy of 84.4 %.

Accuracy for each subgroup is expressed as the ratio of number of correctly classified cases within the subgroup over total number of cases in the subgroup. Total classification accuracy shows the overall performance of the neural network. The MLPNN classified the subgroups of primary generalised epilepsy with the accuracy of 84.4 %. It classified generalized tonik-klonik, absans, myoclonic and more than one type seizures epilepsy group correctly with accuracy of 78.5 %, 80 %, 50 % and 91.6 % respectively. These outcomes indicate that this model may classify the subgroups of primary generalised epilepsy successfully after it is developed.

5 Conclusion and Future Work

This paper briefly introduces the various SC techniques and presents miscellaneous applications in clinical neurology related problems. SC models have been researched and implemented in neurology for a very long time. Recent developments in medicine show that diagnostic expert systems can help physicians make a definitive diagnosis. Automated diagnostic systems are important applications of pattern recognition, aiming at assisting physicians in making diagnostics decisions. Automated diagnostic systems have been applied to a variety of medical procedures, such as electromyogram, electroencephalography. This paper presents applications of soft computing models of the cutting edge researches in neurology domain, specifically for EMG and EEG signals. The presented outcomes indicate that hybrid soft computing models may very successfully be used in different neurology areas such as classification of neurological disorders, diagnosing and treatment determination.

This paper only indicates some researches based on hybrid soft computing and expert and decision support systems. Also, researches on implementation of different artificial intelligence techniques – hybrid soft computing methods can be applied to almost all medical domains, neurology included.

References

1. Zadeh, L.: Soft Computing and Fuzzy Logic. Computer Journal of IEEE Software 11(6), 48–56 (1994)
2. Locatelli, M., Gambini, O., Colombo, C., Beltrami, M., Scarone, S.: A Statistical Approach to Computerized EEG: Preliminary Data on Control Subjects and Epileptic Patients. Brain Topography 3(4), 401–406 (1991)
3. Marinus, J., Visser, M., Jenkinson, C., Stiggelbout, A.M.: Evaluation of the Dutch Version of the Parkinson's Disease Questionnaire 39. Parkinsonism Related Disorder 14(1), 24–27 (2008)
4. Ritter, M.A., Poeplau, T., Schaefer, A., Kloska, S.P., Dziewas, R., Ringelstein, E.B., Heindel, W., Nabavi, D.G.: CT Angiography in Acute Stroke: Does it Provide Additional Information on Occurrence of Infarction and Functional Outcome After 3 Months. Cerebrovascular Disease 22(5-6), 362–367 (2006)
5. Unalan, D., Soyuer, F., Ozturk, A., Mistik, S.: Comparison of 36-item Short-Form Health Survey (SF-36) and World Health Organisation Quality of Life Assessment in Patients with Stroke. Neurology India 56(4), 426–432 (2008)

6. Bugalho, P., Alves, L.: Normal-pressure Hydrocephalus: White Matter Lesions Correlate Negatively with Gait Improvement After Lumbar Puncture. Clinical Neurology and Neurosurgery 109(9), 774–778 (2007)
7. Kincaid, J.C., Prince, K.L., Jimenez, M.C., Skljarevski, V.: Correlation of Vibratory Quantitative Sensory Testing and Nerve Conduction Studies in Patients with Diabetes. Muscle & Nerve 36(6), 821–827 (2007)
8. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid Learning Machines. Neurocomputing 72(13-15), 2729–2730 (2009)
9. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid Intelligent Algorithms and Applications. Information Science 180(14), 2633–2634 (2010)
10. 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)
11. 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(LNAI), vol. 5572, pp. 557–564. Springer, Heidelberg (2009)
12. Ross, T.J.: Fuzzy Logic with Engineering Applications, 3rd edn. John Wiley and Sons, West Sussex (2010)
13. Mendel, J.M.: Uncertain rule-based fuzzy logic systems: Introduction and new direction. Prentice-Hall, Englewood Cliffs (2001)
14. Güler, N.F., Koçer, S.: Classification of EMG signals using PCA and FFT. Journal of Medical Systems 29(3), 241–250 (2005)
15. Oĝulata, S.N., Şahin, C., Erol, R.: Neural network-based computer-aided diagnosis in Classification of primary generalised epilepsy by EEG signals. Journal of Medical Systems 33, 107–112 (2009)
16. Aslan, K., Bozdemir, H., Şahin, C., Oĝulata, S.: Can Neural Network Able to Estimate the Prognosis of Epilepsy Patients Accorrding to Risk Factors? Journal of Medical Systems 34, 541–550 (2010)
17. Ilbay, K.: Übeyli, E. D., Ilbay, G.: Recurent neural networks for diagnosis of Carpal tunnel syndrome using electrophysiologic findings. Journal of Medical Systems 34, 643–650 (2010)
18. Smitha, S.L., Timmisa, J.: An immune network inspired evolutionary algorithm for the diagnosis of Parkinson's disease. BioSystems 94, 34–46 (2008)
19. Kiefer, C., Brockhaus, L., Cattapan-Ludewig, K., Ballinari, P., Burren, Y., Schroth, G., Wiest, R.: Multi-parametric classification of Alzheimer's disease and mild cognitive impairment: the impact of quantitative megnetization transfer MR imaging. Neuroimiging 48(4), 657–667 (2009)
20. Simić, S., Simić, D., Slankamenac, P., Simić-Ivkov, M.: Computer-Assisted Diagnosis of Primary Headaches. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS (LNAI), vol. 5271, pp. 314–321. Springer, Heidelberg (2008)
21. Simić, S., Simić, D., Slankamenac, P., Simić-Ivkov, M.: Rule-Based Fuzzy Logic System for Diagnosing Migraine. In: Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A. (eds.) SETN 2008. LNCS (LNAI), vol. 5138, pp. 383–388. Springer, Heidelberg (2008)

A Hybrid System for Dental Milling Parameters Optimisation

Vicente Vera[1], Javier Sedano[2], Emilio Corchado[3], Raquel Redondo[4], Beatriz Hernando[1], Monica Camara[2], Amer Laham[3], and Alvaro Enrique Garcia[1]

[1] Facultad de Odontología, UCM, Madrid, Spain
{vicentevera,aegarcia}@odon.ucm.es
[2] Dept. of A.I. & Applied Electronics, Castilla y León Technological Institute, Burgos, Spain
{javier.sedano,monica.camara}@itcl.es
[3] Departamento de Informática y Automática, Universidad de Salamanca, Salamanca, Spain
escorchado@usal.es
[4] Department of Civil Engineering, University of Burgos, Burgos, Spain
rredondo@ubu.es

Abstract. This study presents a novel hybrid intelligent system which focuses on the optimisation of machine parameters for dental milling purposes based on the following phases. Firstly, an unsupervised neural model extracts the internal structure of a data set describing the model and also the relevant features of the data set which represents the system. Secondly, the dynamic system performance of different variables is specifically modelled using a supervised neural model and identification techniques from relevant features of the data set. This model constitutes the goal function of the production process. Finally, a genetic algorithm is used to optimise the machine parameters from a non parametric fitness function. The reliability of the proposed novel hybrid system is validated with a real industrial use case, based on the optimisation of a high-precision machining centre with five axes for dental milling purposes.

1 Introduction

The optimisation process of machine parameters could significantly help to increase companies' efficiencies and substantially contributes to costs reductions in preparation and setting machines processes and it also helps in the production process using new materials.

Nevertheless, the variables and parameters setting processes are a well-known problem that has not been fully resolved yet. Several different techniques are proposed in the literature. In [1], is used a Taguchi orthogonal array to optimise effect of injection parameters. In [2] the influence of operating parameters of ultrasonic machining is studied using Taguchi and F-test method. In [3] is researched as to improve the quality of the KrF excimer laser micromachining of metal using the orthogonal array-based experimental design method.

Conventional methods can be greatly improved through the application of soft computing techniques [4].

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 437–446, 2011.

The novel proposed method was tested and validated using a four-step procedure based on several soft computing techniques as artificial neural networks (ANN) and genetic algorithms (GA). Firstly, the dataset is analysed using projection methods such as Principal Component Analysis (PCA) [5], [6], [7] and Cooperative Maximum-Likelihood Hebbian Learning (CMLHL)[8] to analyse the internal structure of the dataset to establish whether the data set is sufficiently informative. Then those methods are applied to perform feature selection as a pre-processing step. It means that if the initial collected data set, once analysed shows a certain degree of clustering, it can be seen as a sign of a representative data set (this means that there is not a single problem related to any sensor when collecting the information and the process is well defined by such data set. Then, the following steps of the process can be applied. And thus the most representative features are identified and used in the following steps. At this phase, a model is generated during the modelling stage to estimate production time errors by modelling techniques. Finally, the ANN model obtained in the last step is used as fitness function to be optimised in the genetic algorithm.

The rest of this paper is organised as follows. Section 2 introduces the unsupervised neural models for analysing the internal structure of the data sets and to perform feature selection.. Section 3 deals with system identification techniques used in the system modelling. Section 4 introduces the applied GA. Section 5 describes the real industrial use case. The final section presents the different models that are used to solve the high precision dental milling optimisation use case. Finally conclusions are set out and some comments on future research lines are outlined.

2 Soft Computing for Data Structure Analysis

Soft Computing is a set of several technologies whose aim is to solve inexact and complex problems [9]. It investigates, simulates, and analyses very complex issues and phenomena in order to solve real-world problems [10]. Soft Computing has been successfully applied in feature selection, and plenty of algorithms are reported in the literature [11], [12], [13].

Feature Selection and extraction [14], [15] entails feature construction, space dimensionality reduction, sparse representations and feature selection among others. They are all commonly used pre-processing tools in machine learning tasks, which include pattern recognition. Although researchers have grappled with such problems for many years, renewed interest has recently surfaced in feature extraction.

In this research, an extension of a neural PCA version [5], [6], [7] and other extensions are used to study the internal structure in the data set as well as to select the most relevant input features for feature selections porpuses.

Then, this research uses the feature selection approach based on the dimension reduction issue. Initially, some projection methods as PCA [5], [6], [7], MLHL [16] and CMLHL [8] are applied. In a first step they aim to analyse the internal structure of a representative data set of a real use case. If after applying these models, a clear internal structure can be identified, this means that the data recorded is informative enough. Otherwise, data must be properly collected again [17], [18].

3 System Modelling Using Identification Algorithms

System identification (SI) [19] aims to obtain mathematical models to estimate the behaviours of a physical process whose dynamic equations are unknown. The identification criterion consists in evaluating the group of candidate models that best describes the dataset gathered for the experiment. The goal is to obtain a model that meets the following premise [19]: a good model is one that makes good predictions and which produces small errors when the observed data is applied.

Classic SI refers to the parametrical literature, which has its origin in the linear system analysis [20]. Nevertheless, increased computational capability and the availability of soft computing techniques have widened research into SI. ANNs are one of the most interesting soft computing paradigms used in SI.

The SI procedure comprises several steps [19], [21]: the selection of the models and their structure, the learning methods [22], [23], [20], the identification and optimisation criteria and the validation method. Validation ensures that the selected model meets the necessary conditions for estimation and prediction. Typically, validation is carried out using three different methods: the residual analysis -by means of a correlation test between inputs, their residuals and their combinations-; the mean squared error (MSE) and the generalisation error value -normalised sum of squared errors (NSSE) - and finally a graphical comparison between the desired outputs and the model outcomes through simulation [20], [17], [18].

4 Genetic Algorithm for System Optimisation

Metaheuristic algorithms are considered as a computational method that optimises a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Metaheuristics are more effective and specialised than the classical heuristics. They combine more exclusive neighbourhood search, memory structures and recombination of solutions and tend to provide better results. However, their running time is unknown and they are usually more time consuming than the classical heuristics. Metaheuristics make few or no assumptions about the problem being optimised and can search very large spaces of candidate solutions. Within these algorithms, there are two well-known types among others, such as the genetic algorithms [24], and the simulated annealing algorithm [25].

GA are adaptive heuristic search algorithm that mimics the process of natural evolution -Darwin's theory about evolution-. This heuristic is routinely used to generate useful solutions to optimisation and search problems. It solves both constrained and unconstrained optimisation problems. GA is a method for moving from one population of “chromosomes” to a new population by using a kind of “natural selection” together with the genetics, inspired operators of crossover, mutation and inversion. In the literature are found a large number of examples [26], [27], [28], [29], [30].

5 A High Precision Industrial Use Case Scenario

This study describes the way in which a hybrid artificial intelligent system can be applied to improve the last step of a high precision industrial system for the manufacture of metal dental pieces, by optimising the time error detection for dental milling process, as shown in Figure 1.

Fig. 1. Metal pieces manufactured by a dynamic high-precision machining centre with five axes

A dynamic high-precision machining centre with five axes was applied in this research. This real industrial use case is described by an initial data set of 109 samples obtained by a dental scanner in the manufacturing of dental pieces with a toric tool characterized by 7 input variables (Radius, Number of pieces, Thickness, Revolutions, Feed rate X, Y and Z) and 1 output variable -Real time of work- as shown in Table 1. Time errors for manufacturing are the difference between the estimated time by the machine itself and real production time -negative values indicates that real exceeds estimated time-.

Table 1. Values of each variable used in the process

Variable (Units)	Range of values
Radius (mm.)	0.25 to 2
Number of pieces	1 to 4
Thickness (mm.)	8 to 18
Revolutions per minute (RPM)	10,00 to 38,000
Feed rate X	75 to 3,000
Feed rate Y	75 to 3,000
Feed rate Z	75 to 2,000
Real time of work (s.)	81 to 1,924
Time errors for manufacturing (s.)	-3 to -332

6 Optimising a Real Dental Milling Process

The manufacturing of dental pieces process optimisation in terms of time errors based on the optimisation of the system behaviour is carried out by means of an ANN estimated model. Firstly, the dental manufacturing process is parameterised and its dynamic performance in normal operation is obtained by the real manufacturing of dental pieces. Then, the gathered data is processed using CMLHL to identify internal data set structures in order to determine the ability of the data set to be modelled and to identify the most relevant features. This allows a third step, knowing a priori, that the model to be obtained can be achieved.

Once the model has been obtained –in the third step-, it is then used as a reference model and also as fitness functions in a GA. The GA calculates the best conditions under normal operating conditions in a dental milling process for manufacturing dental pieces, so if the operator wants to make a dental piece, the best machining conditions might be determined to minimize manufacturing time errors compared to the estimated manufacturing time which is given by the machine itself.

This section deals with the description of each step once the data set is collected (see Section 4). In the next subsection, the generation of the data set which will be used in the process is described. Sub-Section 6.1 presents the PCA and CMLHL steps, in Sub-Section 6.2 the procedure to obtain the time error model is detailed, while in Sub-Section 6.3 the GA is applied.

6.1 Identification of the Relevant Features

PCA and CMLHL are techniques for identifying the internal structure of a data set and also to identify the most relevant variables, as detailed in Section 2. Both of which were applied to this real industrial use case. Then, by means of projection methods it is analysed whether the data set is sufficiently representative of a case study, and the most relevant variables are identified to reduce the computational cost in the third step.

6.2 Modelling a Normal Dental Milling Operation

Once the relevant variables and their transformations have been extracted from the production data, then a model to fit the normal manufacturing operation should be obtained. This is done to identify bias in the estimated production time, which, in the end, is used as fitness function -time error in the manufacturing of dental pieces-. The different model learning methods used in this study were implemented in Matlab© [31].

Moreover, several different indexes were used to validate the models [17], [18] such as the percentage representation of the estimated model; the graphical representation for the prediction - $\hat{y}_1(t \mid m)$ - versus the measured output - $\hat{y}_1(t)$ -; the loss function or error function (V) and the generalization error value.

The percentage representation of the estimated model is calculated as the normalised mean error for the prediction (FIT1). The loss function or error function (V) is the numeric value of the mean square error (MSE) that is computed with the estimation data set. Finally, the generalisation error value is the numeric value of the normalised sum of square errors (NSSE) that is computed alongside with the validation data set (NSSE1) and with the test data set (NSSE2) [32].

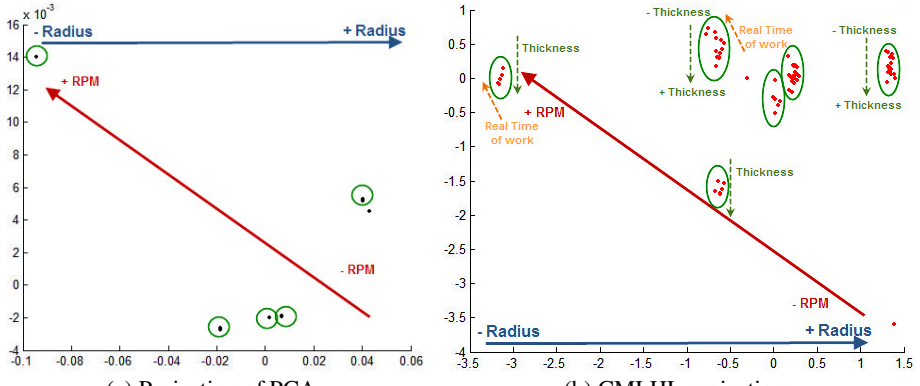
6.3 Optimisation of a Normal Dental Milling Operation

In this case study of dental pieces manufacturing, GAs are concerned with obtaining variables that best optimised the time errors. Firstly this process of optimisation is started with a set of solutions randomly called population –chromosomes-. Then, each individual in the population is evaluated by the fitness function obtained in the last step –ANN model of the manufacturing system-. GA, so as the different types of genetic operators -selection, crossover, mutation- used in this study were implemented in Matlab©.

7 Experiment and Results

The real industrial use case was analysed in order to select the features that best describe the relationships with manufacturing time errors.

CMLHL is a powerful technique for identifying internal dataset structures. The axes forming the projections (Figure 2.a and Figure2.b) represent combinations of the variables contained in the original datasets. In the case of PCA, the model is looking for those directions with the biggest variance, when CMLHL is looking for those which measure how interesting is a dimension/direction. In this case, those are the directions which are as less Gaussian as possible, (by analysing the kurtosis) [8], [16].



(a) Projection of PCA (b) CMLHL projections

Fig. 2. PCA projections (Figure 2.a) and CMLHL projections (Figure 2.b)

As seen in Figure 2, PCA (Figure 2.a) and CMLHL (Figure 2.b), both found a clear internal structure in the dataset. Both methods identified ‘radius’ and 'RPM' variables as the relevant ones. CMLHL projection gives more information since it recognises the 'thickness' as another important variable and also 'real time of work'. CMLHL provides a more sparse representation than the PCA model.

An analysis of the results obtained with the CMLHL model, (Figure 2.b), leads to the conclusion that it has identified several different clusters ordered by 'radius' and 'RPM' variables.

Inside each cluster (Figure 2.b), there are further classifications by ‘thickness’ and 'real time of work'. The dataset can be said to have an interesting internal structure.

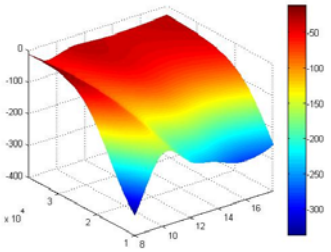
When the dataset is considered sufficiently informative, as in this case, the step for modelling the relations between inputs and production time errors in the process begins, through the application of several conventional ANN modelling systems.

Thus, an ANN was used to monitor time error detection in the manufacturing of dental pieces by using the pre-processed data set from the input and output normalisation step –zero mean and unity standard deviation-, the reduction of the input vectors dimension –the data set gathered in the previous step- and the use of early stopping and Bayesian regularization techniques [33].

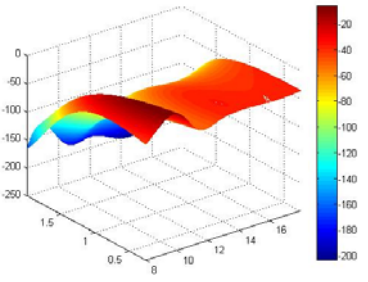
The model was obtained using the Bayesian regularised criterion when the ANN is determined the last step starts and it is then used as fitness functions in a GA. The ANN structure -Feedforward Network- has 25 hyperbolic tangent units -layer 1-, 20 hidden hyperbolic tangent units -layer 2-, 4 hidden hyperbolic tangent units -layer 3- and 1 linear output unit. The network is estimated using the Lenvenberg-Marquardt method. Indexes of the model are FIT1: 80.1%, V: 0.043 and NSSE1: 0.031.

The fitness function is the model of the time error in the dental pieces manufacturing. GA starts with a randomly generated initial population of size 60 individuals. Tournament selection is used to determine the parents for the next generation. Individuals from the current population are selected proportionally to their fitness and forming in this way the basis for the next generation. Two-point crossover combines two parents to form a new individual for the next generation and uniform mutation with a 0.01 rate makes small changes in the individuals in the population. The population obtained by these genetic modifications is evaluated against the fitness function and enters a new search process in the next generation. The algorithm stops after fixed number of generations is reached and the best individual is returned as a solution to the given problem.

Figure 3 shows the output response of the time error for different unnormalised input variable ranges. In Figure 3.a the X-axis shows the thickness, from 8mm to 18 mm., the Y-axis shows the revolutions per minute, from 10,000 to 38,000 in RPM. and the Z-axis represents the unnormalised output variable range from -400 to 0 in s. for a constant value of a radius of 1,5 mm. The time error is shown on the bar, too. In Figure 3.b the X-axis shows the thickness from 8 to 18 in mm, the Y-axis shows the radius, from 0.25 to 2 in mm and the Z-axis represents the unnormalised output variable range from -250 to 0 in s. for a constant value of revolutions per minute of 22,000 RPM. The time error is shown on the bar, too. The time error can be optimised for different values of radius, thickness and revolutions per minute; i.e., it is possible to achieve the less time error and to find the optimal value of the revolutions per minute for a thickness of 14 mm. and a radius of 1.5 mm. -both values fixed- in this case the time error and the revolutions per minute are -38.4 s. and 38,000 RPM., respectively. Also if the radius and the time error are fixed to 1.5 mm. and -100 s., respectively, the thickness and the revolutions to optimise those variables are 12.41 mm. and 19,900 RPM.



(a) 3D graph, the X-axis represents the thickness, the Y-axis the RPM. and the Z-axis the output -time error- for a constant value of radius of 1,5 mm.



(b) 3D graph, the X-axis represents the thickness, the Y-axis the radius, and the Z-axis the output -time error- for a constant value of 22,000 RPM.

Fig. 3. Output response of the time error for different unnormalised input variable ranges

8 Conclusions and Future Work

The novel hybrid artificial intelligence system for the optimisation of this industrial process can be used in the optimisation of machine parameters for industrial processes in general. The process increases the companies' efficiency and substantially contributes to costs reduction of preparation and setting machine processes and it also helps in the production process using new materials. This method has been used in this case for the optimisation and adjustments during the manufacturing process of dental pieces such as implants according to medical specifications of precisely moulded.

The dental milling presents an important time error rate of manufacturing which is about 48%. This is due to the difference between the estimated time by the machine itself and the real production. The model obtained is capable of modelling more than 80% of the actual measurements in relation to time error -modelling more than 90% of the real time of work-. This helps to reduce the error and the variability rate of manufacturing processes down to 10%, compared to 48% initially (acceptable error rate in planning work for dental milling).

Future lines of research include modelling the temperature and the erosion (errors in length or tooth wear), which is a measure of the quality of the dental milling process. Finally, an algorithm will be developed to automatically identify the best operating conditions: minor time errors for the manufacturing of dental pieces and minor erosion. The resulting model would moreover be applied to different metals used in prosthetic dentistry and in other industrial processes.

Acknowledgments. This research is partially supported through projects of the Spanish Ministry of Science and Innovation TIN2010-21272-C02-01, and PID 560300-2009-11], the Spanish Ministry of Science and Technology TIN2008-06681-C06-04] and Junta de Castilla y Leon exp. CCTT/10/BU/0002. The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria, S.A.,

within the framework of project MAGNO2008 - 1028.- CENIT also funded by the Spanish Ministry of Science and Technology and also to ESTUDIO PREVIO (Madrid-Spain) for its collaboration in this research.

References

1. Ozcelik, B., Ozbay, A., Demirbas, E.: Influence of injection parameters and mold materials on mechanical properties of ABS in plastic injection molding. International Communications in Heat and Mass transfer 37(9), 1359–1365 (2010)
2. Kumar, V., Khamba, J.S.: Statistical analysis of experimental parameters in ultrasonic machining of tungsten carbide using the Taguchi approach. Journal of the American Ceramic Society 91(1), 92–96 (2008)
3. Li, J., Ananthasuresh, G.K.: A quality study on the excimer laser micromachining of electro-thermal-compliant micro devices. Journal of Micromechanics and microengineering 11(1), 38–47 (2001)
4. Chang, H.-H., Chen, Y.-K.: Neuro-genetic approach to optimize parameter design of dynamic multiresponse experiments. Applied Soft Computing 11(1), 436–442 (2011)
5. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine 2(6), 559–572 (1901)
6. Hotelling, H.: Analysis of a complex of statistical variables into Principal Components. Journal of Education Psychology 24, 417–444 (1933)
7. Oja, E., Ogawa, H., Wangviwattana, J.: Principal Components Analysis by Homogeneous Neural Networks, part 1, The Weighted Subspace Criterion. IEICE Transaction on Information and Systems E75D, 366–375 (1992)
8. Corchado, E., Fyfe, C.: Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. Int. Journal of Pattern Recognition and Artificial Intelligence 17(8), 1447–1466 (2003)
9. Kohonen, T.: The self-organizing map. Neurocomputing 21(1-3), 1–6 (1998)
10. Sedano, J., de la Cal, E., Curiel, L., Villar, J.R., Corchado, E.: Soft Computing for detecting thermal insulation failures in buildings. In: Proceedings of the 9th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2009, Gijon, Spain (2009)
11. Leray, P., Gallinari, P.: Feature selection with neural networks. Behaviormetrika 26 (1999)
12. Verikas, A., Bacauskiene, M.: Feature selection with neural networks. Pattern Recognition Letters 23(11), 1323–1335 (2002)
13. Diaconis, P., Freedman, D.: Asymptotics of Graphical Projections. The Annals of Statistics 12(3), 793–815 (1984)
14. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research, Special Issue on variable and Feature Selection 3, 1157–1182 (2003)
15. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Knowledge and Data Engineering, IEEE Transactions 17(4), 491–502 (2005)
16. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. Data Mining and Knowledge Discovery 8(3), 203–225 (2004)
17. Vera, V., Corchado, E., Redondo, R., Sedano, J., et al.: Optimizing a dental milling process by means of Soft Computing Techniques. In: 10th International Conference on Intelligent Systems Design and Applications (ISDA 2010), pp. 1430–1435. IEEE, Los Alamitos (2010)

18. Vera, V., Corchado, E., Redondo, R., Sedano, J., et al.: A bio-inspired computational highprecision dental milling system. In: Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC 2010), pp. 430–436. IEEE, Los Alamitos (2010)
19. Ljung, L.: System Identification.Theory for the User, 2nd edn. Prentice-Hall, Upper Saddle River
20. Sedano, J., Corchado, E., Curiel, L., et al.: The application of a two-step AI model to an automated pneumatic drilling process. International Journal of Computer Mathematics 86(10-11), 1769–1777 (2009)
21. Nørgaard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural Networks for Modelling and Control of Dynamic Systems. Springer, London (2000)
22. Stoica, P., Söderström, T.: A useful parametrization for optimal experimental design. IEEE Trans. Automatic. Control AC-27 (1982)
23. He, X., Asada, H.: A new method for identifying orders of input-output models for nonlinear dynamic systems. In: Proc. of the American Control Conf. S.F., California, pp. 2520–2523 (1993)
24. Hayes-Roth, F.: Review of Adaptation in Natural and Artificial Systems. In: Holland, J.h. (ed.) SIGART Bull., pp. 15–15. The u. of michigan press, Ann Arbor (1975)
25. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
26. Aliev, R.A., Aliev, R.R., Guirimov, B., Uyar, K.: Dynamic data mining technique for rules extraction in a process of battery charging. Appl. Soft Comput., 1252–1258 (2008)
27. Chaudhry, I., Drake, P.: Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms. The International Journal of Advanced Manufacturing Technology 42, 581–594 (2009), doi:10.1007/s00170-008-1617-z
28. Fujita, S.: Retrieval parameter optimization using genetic algorithms. Inf. Process. Manage. 45, 664–682 (2009)
29. Oliveira, A.L.I., Braga, P.L., Lima, R.M.F., Cornélio, M.L.: Ga-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. Inf. Softw. Technol. 52, 1155–1166 (2010)
30. Palanisamy, P., Rajendran, I., Shanmugasundaram, S.: Optimization of machining parameters using genetic algorithm and experimental validation for end-milling operations. The International Journal of Advanced Manufacturing Technology 32, 644–655 (2007), doi:10.1007/s00170-005-0384-3
31. Demuth, H., Beale, M., Hagan, M.: Neural Network Toolbox User's Guide. The Mathworks, Inc. (2010)
32. Sedano, J., Corchado, E., Curiel, L., Villar, J.R., de la Cal, E.: Detection of heat flux failures in building using a soft computing diagnostic system. Neural network world 20(7), 883–898 (2010)
33. Mackay, D.J.C.: Bayesian interpolation. Neural Computation 4(3), 415–447 (1992)

A Hybrid Color Distance for Image Segmentation

R. Moreno, M. Graña, and A. d'Anjou

Computational Intelligence Group, Universidad del País Vasco
http://www.ehu.es/ccwintco

Abstract. Our hybrid color distance is inspired in the human vision system, simulating the sensitivity to intensity and chromaticity of the the retina's cells: cones and rods. This approach provides excellent edge detection and is the core of our method. The segmentation output depends on the hybrid distance parameters, whose values define the segmentation method balance between intensity and chromaticity preference.

Keywords: Hybrid color distance, Spherical coordinates, Dichromatic reflection model, Image segmentation.

1 Introduction

Image segmentation is one of the key topics in image processing and computer vision, specially in robotics applications [1]. It is the first step of a lot of computer vision systems and the result quality of the global process depends on it. We find in the literature methods based in watershed [2,3], in clustering process [4], and other features [5]. The watershed methods are based on the gradient information, usually this gradient is obtained from the image intensity, and the chromatic information is ignored. The clustering processes are completely dependent of the color space, therefore classes with few pixels can be incorrectly classified. The segmentation result is a labeled image where each pixel belong to a class, and a class is a connected region in the image.

Uncontrolled illumination, noise and edge detection are problems strongly related. It is most important the use of a suitable distance. We introduce a hybrid distance. It is grounded in the dichromatic reflection model (DRM) [6] formulated in a spherical interpretation of the RGB color space, it helps to avoid the shadows and shines effect. Besides, on one hand with this distance we can parametrize noise tolerance and on the other hand we can adapt this distance for a optimal edge detection. We look for an algorithm with a good behavior over shines and shadows in the image, good noise tolerance, and finally and the most important, it must give us the "desired perceptual result". Perceptual quality is difficult to obtain because it tries to mimic a human mind ability poorly defined in computational terms. According to [7], "the image segmentation problem is basically one of psycho-physical perception, and therefore not susceptible to a purely analytical solution". Tuning the proposed hybrid distance parameters, we can configure the segmentation method behavior, therefore matching the result with our "desired perceptual output".

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 447–454, 2011.

This paper is outlined as follows. In Sec.2 we will introduce the spherical interpretation of the RGB color space, its correspondence with chromaticity and its relation with DRM. Sec.3 explains the hybrid distance. Sec.4 describes our method. In Sec.5 we will present our experimental results. Finally Sec.6 gives the conclusions.

2 Spherical Coordinates and Chromaticity

There are several works that use a spherical interpretation of RGB color space looking for photometric invariants [8,9]. We are interested in the correspondence between the angular parameters (θ, ϕ) an the chromaticity Ψ.

An image pixel's color corresponds to a point in the RGB color space $\mathbf{c} = \{R_\mathbf{c}, G_\mathbf{c}, B_\mathbf{c}\}$. The vector going from the origin up to this point can be represented using spherical coordinates $\mathbf{c} = \{\theta_\mathbf{c}, \phi_\mathbf{c}, l_\mathbf{c}\}$, where θ is zenithal angle, ϕ is the azimuthal angle and l is the vector's magnitude. In the RGB color space, chromaticity $\Psi_\mathbf{c}$ of a color point is represented by its normalized coordinates $r_\mathbf{c} = \frac{R_c}{R_\mathbf{c}+G_\mathbf{c}+B_\mathbf{c}}$, $g_\mathbf{c} = \frac{G_c}{R_\mathbf{c}+G_\mathbf{c}+B_\mathbf{c}}$, $b_\mathbf{c} = \frac{B_\mathbf{c}}{R_\mathbf{c}+G_\mathbf{c}+B_\mathbf{c}}$, such that $r_\mathbf{c} + g_\mathbf{c} + b_\mathbf{c} = 1$. That is, chromaticity corresponds to the projection on the chromatic plane Π_Ψ, defined by the collection of vertices of the RGB cube $\{(1,0,0),(0,1,0),(0,0,1)\}$, along the line defined as $L_\mathbf{c} = \{y = k{\cdot}\Psi_\mathbf{c}; k \in \mathbb{R}\}$. In other words, all the points in line $L_\mathbf{c}$ have the same chromaticity Ψ_c, which is a 2D representation equivalent to one provided by the zenithal and azimuthal angle components of the spherical coordinate representation of the a color point. Given an image $\mathbf{I}(\mathbf{x}) = \left\{(R,G,B)_x ; \mathbf{x} \in \mathbb{N}^2\right\}$, where $\mathbf{x}$ refers to the pixel coordinates in the image grid domain, we denote the corresponding spherical representation as $\mathbf{P}(\mathbf{x}) = \left\{(\phi,\theta,l)_x; \mathbf{x} \in \mathbb{N}^2\right\}$, which allows us to use $(\phi,\theta)_x$ as the chromaticity representation of the pixel's color.

2.1 DRM Expressed in Spherical Coordinates

The Dichromatic Reflection Model explains the perceived color intensity $\mathbf{I} \in \mathbb{R}^3$ of each pixel in the image as addition of two components, one diffuse component $\mathbf{D} \in \mathbb{R}^3$ and a specular component $\mathbf{S} \in \mathbb{R}^3$. The diffuse component refers to the chromatic properties of the observed surface, while the specular component refers to the illumination color.

A scene with several surface colors, the DRM equation must assume that the diffuse component may vary spatially, while the specular component is constant across the image domain. In spherical coordinates is expressed below:

$$\mathbf{I}(\mathbf{x}) = (\theta_\mathbf{D}(\mathbf{x}), \phi_\mathbf{D}(\mathbf{x}), l_\mathbf{D}(\mathbf{x})) + (\theta_\mathbf{S}, \phi_\mathbf{S}, l_\mathbf{S}(\mathbf{x}))$$

For diffuse pixels (it means, pixels without specular component) the zenithal ϕ and azimuthal θ angles are almost constant, while they are changing for specular pixels (it means, pixels with specular component), and dramatically changing among diffuse pixels belonging to different color regions. Therefore, the angle between the vectors representing two neighboring pixels $\mathbf{I}(x_p)$ and $\mathbf{I}(x_q)$, denoted

$\angle(I_p, I_q)$, reflects the chromatic variation among them. For two diffuse pixels in the same chromatic regions, this angle must be $\angle(I_p, I_q) = 0$ because they will be collinear in RGB space. Diffuse pixels are much more abundant than specular pixels, for this reason spherical coordinates is the best way to identify the true surface chromaticity. In diffuse regions the angular parameters are more or less constant. We will use this property for the image segmentation.

3 Edge and Distance

An edge appears in the image when two neighboring pixels have different properties. There are several approaches to implement edge detection by convolution of derivative kernels like Sobel or Prewitt over the image intensity. The direct consequence of this approach is a bad color edge detection because the chromatic information is ignored. Improving these first approach, its convolution masks can be applied to the color using an euclidean color distance, or better using hybrid distances [10]. The core of all edge detection methods are the definition of a distance. Edge measurement is obtained computing this distance among neighboring pixels.

In the DRM formulation, the diffuse component is better characterized by the angular components (θ, ϕ). They are mostly uniform in regions where pixels have close chromatic properties, and it is independent of the intensity, therefore independent of the illumination, assuming an uniform chromatic illumination. In a limit case, if we use only angles to detect edges black-white borders may go undetected because both colors fall in the same achromatic line, and hence they have the same chromaticity. We formulate a hybrid distance to solve this problem.

In images capturing the visual content depend on the contrast between colors and also on the intensity contrast. If we work with only one of these features we lose a lot of important information. Then the next question is, what is more important, intensity or chromaticity? and in what proportion?

Taking inspiration in the human vision, in regions with poor illumination it is better to use intensity. However in regions with good illumination, chromaticity is the main feature for the color detector. Two neighboring image regions can have different colors with the same light intensity, therefore the edge between them may go undetected using only intensity. On the other hand, achromatic surfaces are very abundant in nature and the only way to difference colors in the black-white interval is using the intensity differences. After these considerations we can see that it is necessary to work with chromaticity and intensity. But the use of both together through a distance in the RGB color space (for example the Euclidean distance) is a bad idea, because the euclidean distance between two points near to the origin is very different than between two points in the opposite corner, because neither intensity distance nor chromaticity distance are based on the three-dimensional euclidean distance. Coming back to the human vision, in the retina we have two kind of photoreceptor cells; rods and cones. The first one is an intensity detector and the other one is a chromatic detector. Both need

different energy for its activation. Rods need few energy, for this reason with poor illumination human beings can detect intensity differences. Cones needs more energy and for this reason colors are detected only with a good illumination. Intensity is most important in dark regions and chromaticity importance grows when the illumination is better. This transition is expressed with the graph showed in the Fig. 1. This function represents the chromatic activation function. For values below a it is inactive, for values between a and b it goes from its minimum energy to its maximum energy h following a sinusoidal shape. Finally for values bigger that b its energy is always h. The three parameters a, b, h are in the range $[0, 1]$. The region under the green line is the chromatic importance and its complementary, the region over this line is the intensity importance.

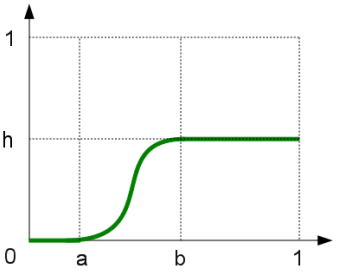


Fig. 1. Chromatic activation function α

The function $\alpha(I)$ depends of the image intensity I.Its mathematical expression is as follows:

$$\alpha(I) = \begin{cases} 0 & I \leq a \\ \frac{h}{2} + \cos\left(\frac{(I-a)\cdot\pi}{b-a} + \pi\right) & a < I < b \\ h & I \geq b \end{cases} \tag{1}$$

where i depends on the intensity. To apply this distance to two colors we compute $I = |l_{\mathbf{c}_1} - l_{\mathbf{c}_2}|/2$ where $l_{\mathbf{c}_1}, l_{\mathbf{c}_2}$ are the intensity component l of the spherical coordinates of the colors $\mathbf{c}_1, \mathbf{c}_2$ and we can express it as $\alpha(\mathbf{c}_1, \mathbf{c}_2)$.

Now we can formulate an hybrid distance between any two colors $\mathbf{c}_1, \mathbf{c}_2$ as follows:

$$d_H(\mathbf{c}_1, \mathbf{c}_2) = (1 - \alpha(\mathbf{c}_1, \mathbf{c}_2)) \cdot d_I(\mathbf{c}_1, \mathbf{c}_2) + \alpha(\mathbf{c}_1, \mathbf{c}_2) \cdot d_C(\mathbf{c}_1, \mathbf{c}_2) \tag{2}$$

where d_I is an intensity distance as $d_I(\mathbf{c}_1, \mathbf{c}_2) = |l_{\mathbf{c}_1} - l_{\mathbf{c}_2}|$ and d_C is a chromatic distance as $d_C(\mathbf{c}_1, \mathbf{c}_2) = \sqrt{(\theta_{\mathbf{c}_1} - \theta_{\mathbf{c}_2})^2 + (\phi_{\mathbf{c}_1} - \phi_{\mathbf{c}_2})^2}$.

4 Segmentation

We borrow from [11] a formal definition of image segmentation: If P() is a homogeneity predicate defined on groups of connected pixels, then segmentation is a

partition of the image domain set F into connected subsets or regions $(S_1, S_2, ..., S_n)$ such that $\bigcup_{i=1}^{n} S_i = F$ with $\forall i \neq j, S_i \cap S_j = \varnothing$.

To obtain the partition in the aforegoing definition, we label all image pixels. Each label corresponds to a chromaticity value. Our algorithm output is a bi-dimensional matrix of integer labels, where each label identifies one and only one connected region in the image. Therefore, if there exist two separated regions with the same color, the segmentation output will represent this two regions by two different labels.

This segmentation method is based is the proposed hybrid distance. The algorithm examines all the pixels in sequence, assigning them labels according to the labeling of the pixels in its neighborhood. We consider 8-connectivity, so all the operations refer to the pixels' 8-neighborhood $N_8(p)$. Four parameters configure the algorithm behavior. On one hand the distance parameters a, b, h previously explained. On the other hand a threshold δ to test color similarity. We decide that two colors $\mathbf{c}_1, \mathbf{c}_2$ are equivalent for segmentation purposes if $d_H(\mathbf{c}_1, \mathbf{c}_2) < \delta$. We will call nearest neighbors to the subset of $NN(p) \subseteq N_8(p)$ pixels with equivalent colors.

Pixels in $NN(p)$ may be labeled or not, and, if some of them are labeled, the same or different labels may occur. If all $NN(p)$ pixels have the same label, the current pixel p and all unlabeled pixels in $NN(p)$ will assume this label. When pixels in $NN(p)$ have different labels, it means that current pixel is at a boundary point between some previously labeled regions whose colors are equivalent, and therefore we must merge them in an unique label, this is known as "region merging". Again, all the pixels in $NN(p)$ are labeled according to the result of the region merging.

Contrary to other labeling algorithms, a special feature of this algorithm is that when examining a pixel we can assign labels to more than one pixel. The current pixel is always labeled in the current iteration, however we make profit from current neighborhood information to label pixels that are ahead in the examination procedure. This anticipatory strategy speeds up the process. For each region label we save the mean chromaticity of its pixels and the amount of pixels in the region. The segmentation algorithm keeps a label counter R, and produces a map Ψ_R assigning to each region label its chromatic value and size. Therefore, an image region label is described by $\Psi_R = [R, (\theta, \phi), \#members]$.

5 Experimental Results

The natural output of a segmentation method is a labels' vector, in this case is a bi-dimensional integer matrix, where each number is linked to a label. For a good visual supervision, the output images are drawn using the label's chromaticity and an uniform intensity ($l = 0.7$).

To validate the proposed segmentation method we will experiment with two different kind of images, on one hand the well-know Berkeley image database [12] and on the other hand a private collection of images taken by the robot NAO. Fig. 2 shows results on the Berkeley database and Fig. 3 shows the results on the

Fig. 2. Experimental results using Berkeley database

robot images. The parameter settings for the experiments are: $\delta = 0.02$, $a = 0.2$, $b = 0.4$ and $h = 0.5$. With these values for the dark regions (intensity less than .2) we use only the intensity distance, whereas for pixels with an intensity bigger than .4 we use a hybrid distance where intensity and chromaticity have the same importance. (Images are normalized to the unit interval).

Respect to the first experiment, using the Berkeley database, images are very different each others and we are using the same parameters and as we can see results are goods. However, image segmentation is a human ability, therefore difficult to compute. It means that for images with different properties we can expect different segmentations. To adapt the output algorithm with our wished results we must set the correct parameters. In fact, the estimation of the correct parameters depending of the images properties is an important aspect and it may be studied in next works.

Respect to the second experiment, images are taken in similar illumination conditions, therefore results are more stable than in the first experiment. It is important to realize the good results avoiding shines and detecting correctly regions with different chromatic properties.

Regarding the speed up, it depends on the expected granularity, or in other words, it depends on the output size. In the case of the robot images, to resize the input image is a good idea because robot images are very noisy due to the robot movements and resizing we remove noise and reduce the computational cost. These experiments have been computed in a laptop with a processor Intel

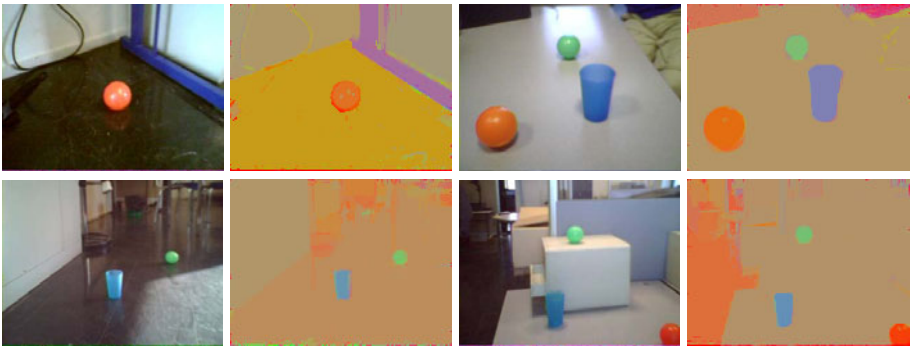


Fig. 3. Robot Images

Core i3 M330 with 4GB of memory, and the code has been written in C#. If the output size is 160x120 it has an average time response of 250ms , and if the output has a 80x60 size of 50ms.

6 Conclusions

In this work we propose a segmentation method starting with a physical approach of the reflectance phenomenon, and hence grounding our method in the dichromatic reflexion model. The spherical coordinate interpretation of the RGB color space is the best way to take profit from the DRM theory in the RGB color space. In spherical coordinate representation it is very easy to detect diffuse regions in the image, however the result is very sensitive to noise in low illumination regions. We explain the advantages of working with intensity and chromaticity in two different ways, and we propose a hybrid distance. It must cover our segmentation expectations through the tuning of its parameters. Finally, to design a segmentation algorithm we apply this distance in a neighborhood-based labeling algorithm.

The introduced method follows a human vision inspiration and it is very versatile thanks to its parametrization and it can be adapted to different expectations of segmentation. As we can see in the experimental results it has a good behavior in shines and shadows. Besides it is fast and can be applied in real time in robotic contexts.

As further works we will to study an automatic estimation of the parameters a, b, h of this method.

References

1. Duro, R.J., Graña, M., de Lope, J.: On the potential contributions of hybrid intelligent approaches to multicomponent robotic system development. Information Sciences 180(14), 2635–2648 (2010)

2. Hamarneh, G., Li, X.: Watershed segmentation using prior shape and appearance knowledge. Image Vision Comput. 27(1-2), 59–68 (2009), http://portal.acm.org/citation.cfm?id=1456803
3. Osma-Ruiz, V., Godino-Llorente, J.I., Saenz-Lechon, N., Gomez-Vilda, P.: An improved watershed algorithm based on efficient computation of shortest paths. Pattern Recognition 40(3), 1078–1090 (2007), http://www.sciencedirect.com/science/article/B6V14-4KNM9X5-1/2/8f3106780d634e52a46388f060a9060f
4. Lézoray, O., Charrier, C.: Color image segmentation using morphological clustering and fusion with automatic scale selection. Pattern Recognition Letters 30(4), 397–406 (2009), http://www.sciencedirect.com/science/article/B6V15-4TYR03B-1/2/edca3e28022072e20f02fb1088c01a34
5. Martin, J.A., de Lope, J., Sanpos, M.: Orthogonal variant moments features in image analysis. Information Sciences 180(6), 846–860 (2010)
6. Shafer, S.A.: Using color to separate reflection components. Color Research and Aplications 10, 43–51 (1984), http://hdl.handle.net/1802/2728
7. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. Pattern Recognition 26(9), 1277–1294 (1993), http://www.sciencedirect.com/science/article/B6V14-48MPPFT-1T0/2/e14ee69e6208152164c7c7791b53b6da
8. Mileva, Y., Bruhn, A., Weickert, J.: Illumination-Robust variational optical flow with photometric invariants. Pattern Recognition, 152–162 (2007), http://dx.doi.org/10.1007/978-3-540-74936-3_16
9. van de Weijer, J., Gevers, T.: Robust optical flow from photometric invariants. In: Proceedings Paper International Conference On Image Processing ICIP 2004, vol. 1–5, pp. 1835–1838. IEEE, Los Alamitos (2004)
10. Moreno, R., Graña, M., Zulueta, E.: RGB colour gradient following colour constancy preservation. Electronics Letters 46(13), 908–910 (2010), http://link.aip.org/link/?ELL/46/908/1
11. Fu, K., Mui, J.: A survey on image segmentation. Pattern Recognition 13(1), 3–16 (1981), http://www.sciencedirect.com/science/article/B6V14-48MPHHJ-7J/2/76e566d2802c76a0990e2a01f68ae68e
12. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision, vol. 2, pp. 416–423 (July 2001)

Empirical Study of Q-Learning Based Elemental Hose Transport Control

Jose Manuel Lopez-Guede, Borja Fernandez-Gauna, Manuel Graña, and Ekaitz Zulueta

Grupo de Inteligencia Computacional, Universidad del Pais Vasco
jm.lopez@ehu.es

Abstract. Non-rigid physical elements attached to robotic systems introduce non-linear dynamics that requires innovative control approaches. This paper describes some of our results applying Q-Learning to learn the control commands to solve a hose transportation problem. The learning process is developed in a simulated environment. Computationally expensive but dynamically accurate Geometrically Exact Dynamic Splines (GEDS) have been used to model the hose to be transported by a single robot, showing the difficulties of controlling flexible elastic passive linking elements.

1 Introduction

Our group has been working in developing suitable control algorithms for Linked MultiComponent Robotic Systems (L-MCRS) as categorized by [1]. This kind of systems offer an inherent complexity due to the dynamics of the physical links and no relevant literature about this subject can be found. This complexity adds on the difficulty of dealing with collections of independent robots [2]. As part of the on-going work, we have started applying different control approaches, some of them based on analytical detailed models [3,4], some using simplified spring-like linking-elements [5]. As a prototypical case we have considered the the hose transportation problem with a single mobile robot at the tip of the hose. This is a simple formulation of the problem, and the results can serve as a starting point for further generalization. We have proposed Q-Learning [6] as the basic approach to learn controllers for this system through experience. Q-Learning [7,8,9,10] is an appealing learning algorithm from the Reinforcement Learning family able to learn from on-line experience without requiring accurate knowledge of the environment and has been successfully used in robotic systems. Given some sensorial data to characterize the world state, an action selection policy selects an action and stores information regarding the quality of the action taken as qualified by a predefined reward system. We have tested several ways of computing this reward, giving different values to the final state when the system ends in a state that can not be labeled as a failure or as having reached the goal. Reinforcement results were provided by the accurate simulation of L-MCRS developed[1] by our group [3,4] based on the Geometrically Exact Dynamic Splines

[1] Available at http://www.ehu.es/ccwintco/index.php/GIC-source-code-free-libre

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 455–462, 2011.

(GEDS) [11] approach to build dynamical model of uni-dimensional objects. In this work, the control points of splines have been used with an additional fourth coordinate to represent the hose-twisting. This paper extends the results giving further insight into the effect of the different learning parameters by the conducted exhaustive experiments.

This paper is organized as follows: section 2 explains the specifics details about the experimental system and the Q-learning strategy. Section 3 gives the results of the different experiments carried out. Our conclusions and comments are given in section 4.

2 Experimental System Design

We will deal with the elemental hose system in this paper, which is composed of one hose segment attached to a fixed end (the source) and whose other end (the tip) is transported by a mobile robot attached to it. The fixed end is set as the middle of the configuration space. The task for the robot is to bring the tip of the hose to a destination. The working space where the tip-of-the-hose robot moves is a square of size 2×2 m^2. The specific definitions of the Q-learning experiment realized are the following:

- State: we have defined the state using three alternative models: $S = (P_r, P_d, i)$, $S = (P_r, P_d, i, c)$ and $S = (P_r, P_d, i, P_1, P_2)$, where
  - $P_r = (x_r, y_r)$ is the actual position of the tip-of-the-hose robot.
  - $P_d = (x_d, y_d)$ is the desired position of the tip-of-the-hose robot, the goal.
  - i is a binary variable that indicates if the line $\overline{P_r P_d}$ intersects the hose. $i = 1$ means that there is an intersection.
  - c is a binary variable that indicates if the box with corners P_r and P_d intersects the hose. $c = 1$ means that there is an intersection,
  - $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are two points of the hose that are uniformly distributed from one end to the other end.
- Working space discretization: we have considered two different discretization steps of 0,5 m. and 0,2 m. This discretization determines the cardinality of the universe of states that we are working with, and it determines the precision of the coordinates of the points P_r, P_d, P_1 and P_2. Our working space is, thus, partitioned into 16 and 100 boxes respectively.
- Final state: the final state can be of three kinds.
  - Goal: the tip of the hose reaches the goal.
  - Failure: the tip of the hose is blocked in its advance by the hose itself.
  - Inconclusive: the simulation ends without reaching any of the aforementioned states.
- Actions: In our problem we can only interact with the scenario using the mobile robot to change the position of the tip-of-the-hose, so the actions are the possible motion directions of the robot. We have chosen a small set of only four actions: $A = \{ North, South, East, West \}$, meaning that the robot will move in this direction for a length equivalent to the size of the resolution box.

Table 1. Reward systems

$r \leftarrow \begin{cases} +1 & if\ goal \\ -1 & if\ failure \\ 0 & else \end{cases}$	$r \leftarrow \begin{cases} +100 & if\ goal \\ -100 & if\ failure \\ f\,(distance) & else \end{cases}$ $f\,(\cdot) = \left(100 - \frac{dist\,in\,cm}{2}\right)$
reward system code: 10	reward system code: 60
$r \leftarrow \begin{cases} +100 & if\ goal \\ -100 & if\ failure \\ 0 & else \end{cases}$	$r \leftarrow \begin{cases} +100 & if\ goal \\ -1000 & if\ failure \\ f\,(dist,\, i,\, c) & else \end{cases}$ $f\,(\cdot) = \left(100 - \frac{distance\,in\,cm}{2}\right) - 10.i - 10.c$
reward system code: 20	reward system code: 70
$r \leftarrow \begin{cases} +100 & if\ goal \\ -1000 & if\ failure \\ f\,(distance,\, i,\, c) & else \end{cases}$ $f\,(\cdot) = -\left(\frac{distance\,in\,cm}{10} + 10.i + 10.c\right)$	$r \leftarrow \begin{cases} +100 & if\ goal \\ -100 & if\ failure \\ f\,(distance,\, i,\, c) & else \end{cases}$ $f\,(\cdot) = -\left(\frac{distance\,in\,cm}{10} + 10.i + 10.c\right)$
reward system code: 30	reward system code: 80
$r \leftarrow \begin{cases} +100 & if\ goal \\ -100 & if\ failure \\ f\,(distance,\, i,\, c) & else \end{cases}$ $f\,(\cdot) = \left(100 - \frac{distance\,in\,cm}{2}\right) - 10.i - 10.c$	$r \leftarrow \begin{cases} -1000 & if\ goal \\ +1000 & if\ failure \\ f\,(distance,\, i) & else \end{cases}$ $f\,(\cdot) = distance\,in\,cm + 100.i$
reward system code: 40	reward system code: 90
$r \leftarrow \begin{cases} +1 & if\ goal \\ 0 & if\ failure \\ 0 & else \end{cases}$	
reward system code: 50	

– Reward system: We have used several reward systems. In the table 1 we present all the reward systems that we have used.
  - Reward systems 10 and 20: both give a positive reward when reaching the goal, negative when failing and nothing if the end state is inconclusive.
  - Reward system 50: only gives positive reward when reaching the goal.

- The remaining reward systems give positive reward when reaching the goal, negative when failing and for the inconclusive states, a function of the actual distance between the hose tip and the goal. In some cases the reward function is also function of the binary variables c and i.

– α: $[0 < \alpha \leq 1]$, as we suppose that we are working in a deterministic environment we can assume that the value of this parameter is 1, so the Q-table update expression is: $Q(s,\, a) \leftarrow r + \gamma\, max_{a'} Q(s',\, a')$.
– γ: $[0 < \gamma \leq 1]$, we have set this value to 0,9.
– Action selection: we have chosen an ϵ-greedy policy, and we have set this value as $0,2$. So we choose the action a with this criterion:

$$a \leftarrow \begin{cases} \underset{a'}{maxQ}\,(s,\, a') & \text{with probability } (1-\epsilon) \\ any\, a' \in A & \text{with probability } \epsilon \end{cases}.$$

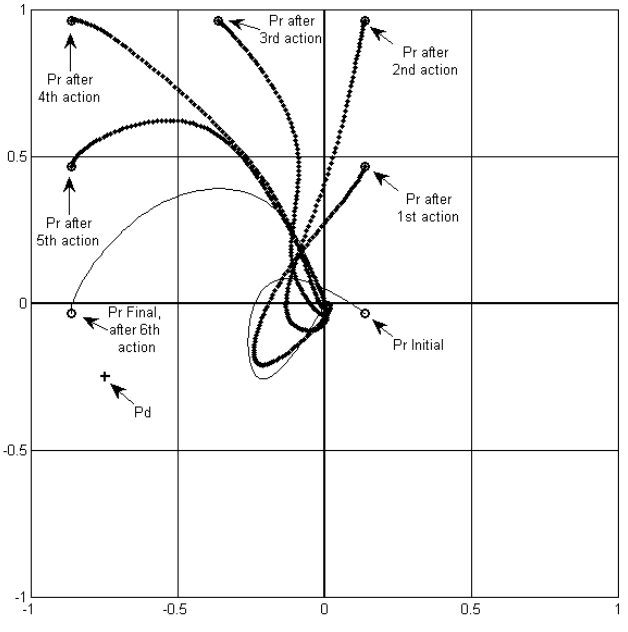


Fig. 1. Episode where the tip of the hose robot reaches the goal

Figure 2 shows a success episode in which the robot carrying the tip of the hose goes from a original position to a destination position.

3 Experimental Results

The experiments that we have designed consisted in the systematic exploration of the combinations of state, reward system and discretization step. Besides,

Table 2. Total episodes of the training phase (thousands of episodes)

	state model				
	$S = (P_y,\ P_d,\ i)$		$S = (P_y,\ P_d,\ i,\ c)$		$S = (P_r,\ P_d,\ i,\ P_1,\ P_2)$
reward system	$\triangle s$ $0'5\,m.$	$\triangle s$ $0'2\,m.$	$\triangle s$ $0'5\,m.$	$\triangle s$ $0'2\,m.$	$\triangle s$ $0'5\,m.$
10	6.410	1.740	5.920	1.410	9.830
20	6.410	460	6.490	370	10.260
30	6.070	1.900	6.700	1.510	12.360
40	4.530	780	4.440	650	12.480
50	7.330	2.260	7.540	1.660	22.620
60	22.650	1.010	20.470	750	
70	23.290	1.090	20.450	620	
80	34.490	2.350	31.270	1.940	
90	37.970	2.810	36.430	1.980	

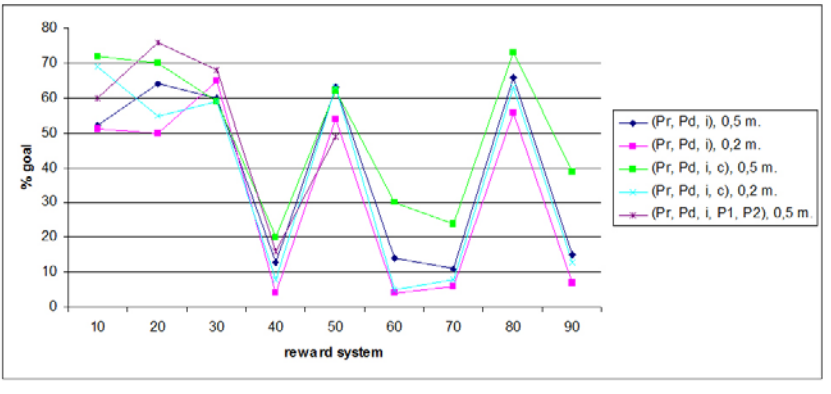


Fig. 2. Percentage of successful runs (reaching goal) obtained in test phase with each reward system and state definition over a hundred simulations per combination of parameters

we have obtained numerical values of the results with different training time (expressed in terms of training episodes) in order to compare the learning of the same systems varying this parameter. The number of total episodes that we have carried out with each combination in the training phase is in table 2. In this way we have obtained the results that are shown in figure 2 and figure 3.

In these figures we show, for each combination of reward system and state model (with each different discretization step), the percentage of episodes where the robot reaches the goal in figure 2, and the percentage of episodes concluded because the maximum allowed step count was reached in figure 3. For each combination we show the results obtained in the test phase with 100 different initial configurations.

The best results correspond to the reward system code 20 with the state defined as $S = (P_r,\ P_d,\ i,\ P_1,\ P_2)$, the success rate, i.e. the percentage of episodes

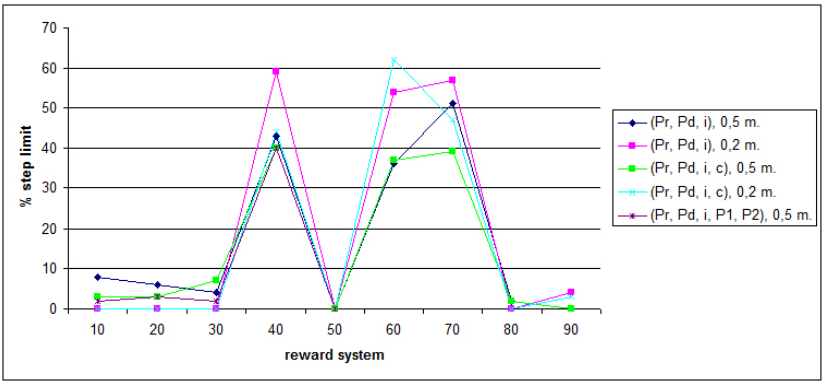


Fig. 3. Percentage of runs terminated because they reached the limit number of steps obtained in test phase with each reward system and state definition over a hundred simulations per combination of parameters

where the robot reaches the goal, is 77% of the test episodes. The 2% of the test episodes concluded because the maximum allowed step count was reached. Finally, 21% of the test episodes failed either because the robot collided with the hose or because the whole system reached a non-feasible position. The worst results come from the reward strategies 40, 60, 70 and 90 that give some combination of the distance to the goal and the binary variables as the reward function in the inconclusive state. They have the worst accuracy results and the higher number of simulations ended because they reached the step limit. The reward systems that gave a null reward or pure negative reward in terms of distance to the goal in the inconclusive final states were the ones with better results, regardless of the definition of the state, which is indicative of the robustness of the approach.

In general, we can see that increasing the training episodes the result in the test phase improve slightly in two ways: firstly, the number of episodes that finish with success have experienced a slight increase. On the other hand, the number of episodes that fail decrease to increase the number of those that finish because the maximum allowed step count was reached.

4 Conclusions

As a matter of interest to our group, we have approached the design of a controller for a hose transportation problem in a L-MCRS multiagent system using Reinforcement Learning methods, more specifically, Q-learning. We have restricted work in this paper to a single robot moving the tip of the hose to a desired position, while the other end is attached to a fixed position. We have tested a number of combinations of the model, the reward systems and the step size used in the discretization process. Results of the training computational experiment are good for some combinations.

In this paper we have paid much attention to the time needed to get a reasonably good training in terms of episodes. Viewing the experimental data we can realize the fact that investing much more time in the training phase, the results improve, but the ratio of that improvement is not linear in relation to the computational effort. As a result of our experiments, we conclude that one of the biggest issue while conducting experiments was the large duration of simulations, mainly because the hose model's computation requirements, but also due to the huge number of episodes needed to explore before learned Q-table exploitation can yield good results.

Further work will be directed to optimize the state-action space representation, while keeping the most important information required for the learning purpose. We will also attempt to apply the learned control strategy on real robots to further validate the results. Long term research will deal with learning control strategies for a collection of robots attached along the hose. We will evaluate the application of hierarchical decomposition techniques [12] and the application of alternative knowledge modeling paradigms such as [13,14].

References

1. Duro, R.J., Graña, M., de Lope, J.: On the potential contributions of hybrid intelligent approaches to multicomponen robotic system development. Information Sciences 180(14), 2635–2648 (2010)
2. Prieto, A., Bellas, F., Caamaño, P., Duro, R.J.: Automatic behavior pattern classification for social robots. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 88–95. Springer, Heidelberg (2010)
3. Echegoyen, Z.: Contributions to visual servoing for legged and linked multicomponent robots. Ph.D. dissertation, UPV/EHU (2009)
4. Echegoyen, Z., Villaverde, I., Moreno, R., Graña, M., d'Anjou, A.: Linked multicomponent mobile robots: modeling, simulation and control. In: Robotics and Autonomous Systems, vol. 58(12), pp. 1292–1305 (2010)
5. Fernandez-Gauna, B., Lopez-Guede, J.M., Zulueta, E.: Linked multicomponent robotic systems: Basic assessment of linking element dynamical effect. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 73–79. Springer, Heidelberg (2010)
6. Fernandez-Gauna, B., Lopez-Guede, J.M., Graña, M.: Learning hose transport control with q-learning. Neural Network World 20(7), 913–923 (2010)
7. Maravall, D., de-Lope, J., Martin, J.A.: Hybridizing evolutionary computation and reinforcement learning for the design of almost universal controllers for autonomous robots. Neurocomputing 72(4-6), 887–894 (2009)
8. Martin, J.A., de Lope, J., Maravall, D.: Adaptation, anticipation and rationality in natural and artificial systems: Computational paradigms mimicking nature. Natural Computing 8(4), 757–775 (2009)
9. Martin, J.A., de Lope, J., Santos, M.: A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers. Neurocomputing 72(13-15), 2806–2814 (2009)
10. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)

11. Theetten, A., Grisoni, L., Andriot, C., Barsky, B.: Geometrically exact dynamic splines. Computer-Aided Design 40(1), 35–48 (2008)
12. Graña, M., Torrealdea, F.J.: Hierarchically structured systems. European Journal of Operational Research 25, 20–26 (1986)
13. Graña, M., d'Anjou, A., Albizuri, F.X., Hernandez, M., Torrealdea, F.J., Gonzalez, A.I.: Experiments of fast learning with high order boltzmann machines. Applied Intelligence 7(4), 287–303 (1997)
14. d'Anjou, A., Graña, M., Torrealdea, F.J., Hernandez, M.: Solving satisfiability via boltzmann machines. IEEE Transactions on pattern analysis and machine intelligence 15, 514–521 (1993)

Towards Concurrent Q-Learning on Linked Multi-Component Robotic Systems

Borja Fernandez-Gauna, Jose Manuel Lopez-Guede, and Manuel Graña

University of the Basque Country (UPV/EHU)

Abstract. When conventional Q-Learning is applied to Multi-Component Robotic Systems (MCRS), increasing the number of components produces an exponential growth of state storage requirements. Modular approaches make the state size growth polynomial on the number of components, making more manageable its representation and manipulation. In this article, we give the first steps towards a modular Q-learning approach to learn the distributed control of a Linked MCRS, which is a specific type of MCRSs in which the individual robots are linked by a passive element. We have chosen a paradigmatic application of this kind of systems: a set of robots carrying the tip of a hose from some initial position to a desired goal. The hose dynamics is simplified to be a distance constraint on the robots positions.

1 Introduction

The key feature that makes traditional control techniques inappropriate for Linked Multi-component Robotic Systems (L-MCRS) [2,3,7] is the non-rigid physical link that imposes certain physical constraints and induces non-linear dynamics in the system. Our previous work deals with the application of distributed control schemes based on consensus techniques to very simple L-MCRS where the links are modeled as springs [5,6]. As an alternative approach to traditional control, we propose [4] the application of *Q-Learning* algorithm to learn from experience.

The *Q-Learning* algorithm is part of the unsupervised *Reinforcement Learning* (RL) [9,11] method family and has become very popular because of its good behavior despite its simplicity. The algorithm does not require any a priori knowledge about the environment and it can even learn from simulated state transitions. A learner is assumed to observe discrete states $s \in S$ from the world, choose a discrete action $a \in A$ to be taken following policy $\pi : S \rightarrow A$ and observe new state s'. A previously defined *reward function* immediately maps perceived states to a scalar real *reward* (r) describing how good or desirable the state is: $r : S \rightarrow \mathbb{R}$. While the *reward* is the immediately observed signal qualifying the observed state, the sum of all the rewards observed over time is called the *value*. Knowledge can be acquired from *episodic* (finite tasks) or *continuous* tasks. Once a reward is obtained, an agent can update its previous knowledge

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 463–470, 2011.

about taking action a in state s (described by a real number $Q(s,a)$) using the following expression:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma * \max_{a'} \{Q(s',a') - Q(s,a)\} \right), \tag{1}$$

where $\alpha \in [0,1]$ is an step-size parameter indicating how fast the agent is desired to learn and $\gamma \in [0,1]$ is a discount-rate parameter that weights earlier rewards higher than later ones. All those *Q-values* are usually stored in a look-up table and each of the entries represents how good an action has been in a state. *Q-Learning* algorithm updates entries as soon as an action is taken based on the estimated value of the new state observed; as it can be seen in equation 1, this estimation is based on the best *Q-value* available in state s'. An action selection algorithm must be defined so an action is selected according to the observed state every time-step. While the straightforward approach (*greedy* action selection) involves always selecting the action with the highest Q-value *exploiting* available knowledge, this prevents the agent *exploring* yet unknown *action-state* pairs and thus, not allowing it to explore possibly better actions. This compromise between *exploration* and *exploitation* is usually solved using a $\epsilon - greedy$ algorithm (a random action is selected with probability ϵ while the best action is chosen with probability $1 - \epsilon$) .

The main drawback of using *Q-Learning* with explicit look-up tables is called *curse of dimensionality*: as the state space increases, the size needed to store the Q matrix grows so fast it easily becomes an unpractical approach to real applications. To deal with this problem in the L-MCRS control learning, we explore the concurrent learning [10] which uses simultaneously multiple RL modules assigning each of them a subtask to carry out. Each module represents the world as a subspace of the sensorial information available to the agent (ideally, it only receives the state input needed to properly learn its subtask) and they all are able to learn concurrently.

Another problem that arises in multi-agent domains is how to assign credit to agents when the team achieves a goal. Three main approaches may be distinguished: *global rewards*, *local rewards* and mixed approaches. Global rewards are those given to the whole team when it achieves a team-level goal. Local rewards are those based exclusively on individual performance. The latter discourages lazy behaviors but doesn't encourage coordination either, so they are not appropriate for team-level goals.

Coordination of agents [1] is also an interesting and non-trivial issue, because if the system does not act in a coordinated way, it may lead to unexpected results. For the scope of this work, we will assume that robots will have permission to move in a round-robin schedule, that is, an explicit order will be defined and robots will make discrete moves in turns so robots can consider the remaining robots as static during their turn.

This paper is structured as follows: first we introduce the *Modular Concurrent Q-Learning* concepts and issues in section 2 including features of our own modular approach to L-MCRS. Section 3 presents the application chosen to test our

approach. Section 4 shows the results obtained giving details of the conducted experiments. Finally, conclusions and future work comments are presented in section 5.

2 Modular Concurrent Q-Learning

Let n be the number of learning agents in a multi-component system, m the number of modules each agent runs and c the number of possible actions each agent can take. For a discretely perceived world state $s \in S$, the i^{th} module is fed with a subset $s_i \epsilon S_i \subset S$ which is expected to be relevant to achieve its goal. Let us denote as $Q_i(s_i, a)$ the Q matrix entry of the i^{th} module relating its partial state s_i and action a.

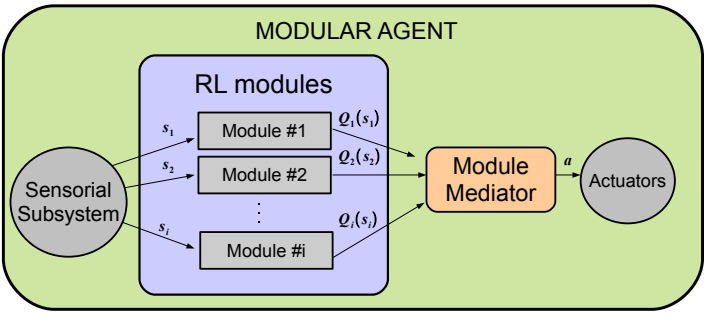


Fig. 1. Modular Q-Learning Scheme

The typical approach to learn different subtasks or behaviors concurrently involves using a *Module Mediator* (also referred to as *Module Arbiter* in the literature) responsible for action selection, as represented in Figure 1. Each module has its own Q matrix representing its partial knowledge of the world state s_i and modules may even compete imposing their preferences to the rest. To select the next action we will follow the Greatest Mass (GM) strategy [12] defined as:

$$\pi(s_i) = \arg\max_{a \in A} \sum_{i=1}^{m} Q_i(s_i, a),$$

which selects the action that maximizes the sum of local agent Q-values.

An additional advantage of using separate modules is that some subtasks (those that can be learnt independently of other agents' policies) can be trained separately, thus reducing the exploration space as the state space shrinks to the module's state subspace $\mid s_i \mid$. This reduction often gives the possibility to consider using *exploring starts* [11], which assure a better exploration of the state space.

Two different type of tasks can be easily distinguished in L-MCRS: those trying to reach one or more goals and those satisfying the physical constraints imposed by the linking element so as to avoid the hard-to-control undesired effects (e.g, force exerted by the hose if stretched). To cope with the different

nature of these two module types, we propose distinguishing goal modules and constraint modules while keeping the structure coherent with that already represented in Figure 1. Constraint modules are expected to learn which actions are not to be carried out in a given state, while goal modules learn how to reach the goal not being even aware of the constraints present. We use positive rewards in the goal-oriented modules whenever the goal is reached, negative rewards in the constraint modules if a constraint is not respected and neutral rewards otherwise. The discount-rate parameter γ is different for both types of learning modules: goal-oriented modules will use a rather low γ parameter and constraint-type modules will work best with a high value, so they can learn on a one step basis.

2.1 Veto-Based Action Selection

To enforce the constraints, we propose to use a simple veto system, allowing constraint modules to impose a veto to actions that have broken constraints in the past. A boolean vector $\boldsymbol{V_i} \in \{true, false\}^c$ is defined for each Module Mediator as follows:

$$V_i(a_j) \begin{cases} true & if\ Q_i(s_i, a_j) < v_t \\ false & else \end{cases},$$

where $i = 1, ..., n$, $j = 1, ..., c$ and v_t is the threshold for imposing the veto. This means that if, under state s_i, a constraint module i has a value $Q_i(s_i, a_j)$ below the veto threshold, action a_j is forbidden for that state. Traditional $\epsilon - greedy$ is then used to select an action among the available ones, assuring this way that taking a random action will not result in an abrupt failure once the modules have a minimum amount of experience, avoiding that way the main problem towards appropriate learning. Instead of manually tweaking the ϵ parameter, we obtain a simple approach to keep goal-focused exploration while avoiding constraint-related termination conditions. Of course, the downside to this technique is that it only works properly with deterministic constraints because exploration is disabled for states found to be undesired in the past.

3 Hose Transport Application

As a paradigmatic application of L-MCRSs, the hose transportation problem has been chosen. A group of n robots attached at fixed points of a hose must carry the tip of the hose to a given position avoiding collisions. The source of the hose is assumed to be located in the $(0, 0)$ cell and robots are identified by an integer number in range $[0, n-1]$, where the robot carrying the tip of the hose is labeled as 0 and the rest labels are given according to the position on the hose as shown in Figure 2. In this figure, $P_i = (P_i^x, P_i^y)$ denotes the discrete coordinates position of the i^{th} robot on the grid representing the world and the goal position will be noted as $G_i = (G^x, G^y)$. For the scope of this article, the hose between i^{th} and $i+1^{th}$ robots is represented as the line segment $P_i - P_{i+1}$,

which has a maximum nominal length of L times the size of grid cells. Because the ultimate goal for the team of robots is for robot number 0 to reach a goal position, no matter where the rest stand as long as constraints are respected, it made sense to use a global reward signal for the goals.

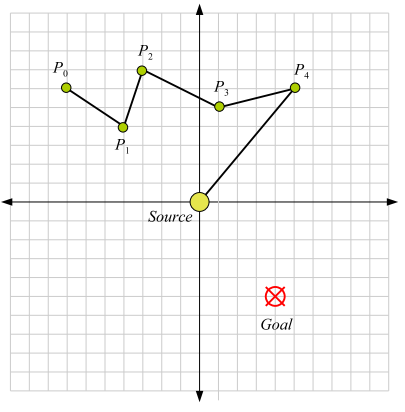


Fig. 2. Representation of the robots and the goal on the grid

3.1 Modules

Our experiments involved a variety of module combinations (including both homogeneous and heterogeneous agents) and the best results were achieved using two goal-modules and three constraint-modules:

1. *Goal-1*: This module models the state as the combination of the distance to the goal and the angle formed by the hose segment and the goal. Training uses a global reward: whenever the tip of the hose reached the goal, a positive reward was given, else a neutral value.
2. *Goal-2*: A simplification modeling the state as the distance to the goal.
3. *Constr-Distances*: The module models the length constraint on the hose segment ahead of the robot. It receives a negative reward every time any of them exceeds the maximum allowed length, and a neutral one otherwise.
4. *Constr-Collisions*: This module learns to avoid collisions between robots and hose segments. State is modeled as a set of boolean values, each of which indicates whether there is an obstacle (robot or hose segment) within one time-step reach in a direction corresponding to one of the allowed actions.
5. *Constr-InGrid:* The robots were desired to stay within a predefined bounds. This module gives a negative reward if the position is outside the allowed bounds, neutral otherwise.

4 Experiments

A set of experiments were conducted to test our modular multi-agent learning approach for L-MCRS. We used a grid of 21×21 cells, a maximum hose length L_{hose} of 26 cell units and the step limit count for episodes was set to 400: episodes that didn't reach the goal within this limit were forced to abort. In all the cases, the ϵ parameter had a fixed value: 0.1. Robots were allowed to take any of 9 actions at each time step: move North, North-east, East, South-east, South, South-west, West, North-west or no move at all. All of them were considered to be deterministic, always reaching the intended position. Episodes were randomly generated to have a more realistic measure of performance, rejecting those that didn't fulfill all the constraints. We report results on the improvement introduced by the use of the veto system (Experiment B) versus the conventional greedy action selection (Experiment A) in the concurrent training of the agents. Two different graphs were plotted for each of the experiments: the success rate of the system related to the constraint module errors for a system with 6 robots, and the success rate for different number of robots.[1]. Videos were generated for each of the experiments to visually validate the simulations[2]. Figure 3 shows the evolution of the percentage of episodes reaching the target as the training evolves for both experiments and for an increasing number of robots. It both cases, increasing the number of robot dramatically reduces the capacity to reach the goal. Hover results are much better for the veto system (figure 3(b)) than for the baseline greedy system (figure 3(a)).

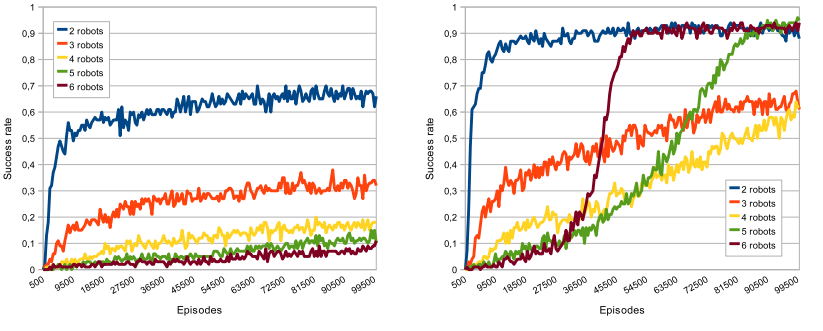


Fig. 3. Experiment A: Success rate for different number of robots. (a) Experiment A, (b) Experiment B.

[1] Note: all charts used express normalized rates $[0, 1]$ on y-axis and episode count on x-axis.

[2] Some videos can be downloaded from `http://www.ehu.es/ccwintco/index.php/Borja-videos`

5 Conclusions

We have presented a modified veto-based version of existing Modular RL techniques suitable to deal with L-MCRS and we have applied it to the multi-robot hose transport problem. Results show that combined use of separate constraint module training and a veto system leads to a very good learning rate and success rate. We also studied the scalability of the system and, although increasing the number of agents decreased slightly the performance of the learning algorithm, results showed very satisfactory.

Our approach has two main limitations: robots are limited to move in turns to avoid the coordination issues, and it assumes deterministic constraints. On the future, our work will focus on introducing the learnt Q matrices in a simulation environment that includes more realistic hose models for further validation of the approach and the assumptions made. It also will address the coordination issues so robots can move simultaneously and will consider completely stochastic environments. More complex tasks and scenarios may also benefit from integrating hierarchical structures and this is also a very appealing field of research.

It is our belief that L-MCRSs have a very unique idiosyncrasy that requires specific approaches and more effort should be focused on this particular research area before more complex applications can be solved. Modular approaches inspired in the general hierarchical systems theory should be pursued [8].

References

1. Busoniu, L., Babuska, R., De Schutter, B.: A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38(2), 156–172 (2008)
2. Duro, R.J., Graña, M., de Lope, J.: On the potential contributions of hybrid intelligent approaches to multicomponen robotic system development. Information Sciences 180(14), 2635–2648 (2010)
3. Echegoyen, Z., Villaverde, I., Moreno, R., Graña, M., d'Anjou, A.: Linked multicomponent mobile robots: modeling, simulation and control. Robotics and Autonomous Systems 58(12), 1292–1305 (2010)
4. Fernandez-Gauna, B., Lopez-Guede, J.M., Zulueta, E., Graña, M.: Learning hose transport control with Q-learning. Neural Network World 20(7), 913–923 (2010)
5. Fernandez-Gauna, B., Lopez-Guede, J.M., Zulueta, E.: Linked multicomponent robotic systems: Basic assessment of linking element dynamical effect. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 73–79. Springer, Heidelberg (2010)
6. Fernandez-Gauna, B., Lopez-Guede, J.M., Zulueta, E., Echegoyen, Z., Graña, M.: Basic results and experiments on robotic multi-agent system for hose deployment and transportation. International Journal of Artificial Intelligence 6(S11), 183–202 (2011)
7. Ferrández, J.M., de la Paz, F., de Lope, J.: Intelligent robotics and neuroscience. Neurocomputing 58(12), 1221–1222 (2010)
8. Graña, M., Torrealdea, F.J.: Hierarchically structured systems. European Journal of Operational Research 25, 20–26 (1986)

9. Maravall, D., de Lope, J., Martin, J.A.: Hybridizing evolutionary computation and reinforcement learning for the design of almost universal controllers for autonomous robots. Neurocomputing 72(4-6), 887–894 (2009)
10. Panait, L., Luke, S.: Cooperative multi-agent learning: The state of the art. Autonomous Agents and Multi-Agent Systems 11(3), 387–434 (2005)
11. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
12. Whitehead, S., Karlsson, J., Tenenberg, J.: Learning multiple goal behavior via task decomposition and dynamic policy merging. In: Robot Learning, pp. 45–78. Kluwer Academic Publishers, Dordrecht (1993)

Evolutionary Procedure for the Progressive Design of Controllers for Collective Behaviors

P. Caamaño, J.A. Becerra, F. Bellas, A. Prieto, and R.J. Duro

Integrated Group for Engineering Research, Universidade da Coruña, Spain
{pcsobrino, ronin, fran, abprieto, richard}@udc.es
http://www.gii.udc.es

Abstract. This paper describes an approach for the progressive construction of controllers for sets of robots performing collective behaviors. The procedure is based on the incremental construction through evolution of a neural multilevel behavior architecture where the higher-level behaviors modulate the actuation of the lower-level ones. This hybridization permits simplifying the design of the behavior controllers and allows obtaining them in evolutionary processes without making the search space huge. From a cognitive point of view, the procedure could be thought of as an incremental learning procedure where the robot first learns basic responses and then uses them within more elaborate decision and actuation processes progressively increasing complexity.

1 Introduction

In the last few years, a lot of effort has been devoted to the generation of coordinated behaviors for large (and sometimes not so large) groups of robots. Work has been carried out on the formalization of the problems [1][2] in order to produce hand crafted algorithms or controllers for particular tasks, as well as on implementation issues [3][4]. However, many of these approaches are particular to a behavior (i.e. foraging, flocking) or environment, often using homogeneous sets of controllers, and cannot be considered a general framework for the generation of collective behaviors. There is, however, some work, such as [5] or [6], that is beginning to appear where some problems are characterized and a study of the production of collective solutions is carried out in order to determine the most appropriate approach for each case. The main problem, which is often ignored, is that the structure of most real world problems does not allow for an easy decomposition into sub-problems or, in robotic terms, particular subtasks, nor are they known beforehand and, consequently, the robot teams that solve them must arise in a joint manner as it is through the interaction of the components doing their tasks and not through the tasks themselves that the final objective is achieved.

In this work we are interested in what could be considered an engineering approach. We want to obtain systems that produce specific behaviors required for particular tasks or processes. We are, thus, contemplating a design process the objective of which is to produce the appropriate controllers for the individual entities participating in the collective process, so that the whole does what it is supposed to do. These types of design processes are necessary in many different realms. Examples can be

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 471–478, 2011.

found for robots in industrial settings, where a few robots must collaborate to perform a task such as exploring an area [3] or for large sets of robots such as in the field of nanotechnology and nanorobotics[7], where thousands of individual entities must be controlled to produce a collective that achieves the goal in a given environment.

As these control systems are complex, the basic idea in this paper for the design of collective behaviors is that one has to provide a coherent controller architecture and construction mechanism that makes efficient use of previously obtained information, and that allows for an automatic incremental construction of the individual behaviors within the collective system [8]. Such mechanism could start from an initial creation of simple low level behaviors, by evolution or learning in simple and controllable circumstances, which are aggregated and modulated by higher level behaviors that are created taking the low level behaviors as primitives (at least, those low level behaviors that are required for the task). Thus, in what follows, we will present a modulation based architecture that is used to produce controllers for multirobot teams that collectively produce behaviors leading to the completion of the task in hand. We will also discuss how it can be implemented through the classical hybridization of evolution and artificial neural network based structures.

2 Description of the Architecture and Procedure

The objective of this procedure is to automaticallyobtain the behavior of a collection of robots acting together making it feasible to reuse previously obtained behaviors,making the production of ever more complex behaviors for therobots more progressive and easier. It is based on a hierarchical modulation based controller behavior architecture where both the controller modules and the connections among them are obtained through evolutionary processes and that is described with detail in [9]. These modules are based on Artificial Neural Networks due to their excellent characteristics as universal function modelers.

Thus, in a first stage, the simplest behaviors (low level behaviors) are identified andobtained separately as a whole using evolutionary processes in very simplified environments. That is, a single ANN is obtained to implement the behavior and take control of effectors. One can have behavior libraries out of which behaviors may be reused and this step can be skipped. Higher-levelcontrollersdo not directly control effectors but modify the way lower level controllers act. They are obtained, together with their connections to lower level controllers, through evolution in environments that are particularized for their particular behavior and make use, if possible, of previously obtained lower level controllers that are appropriate for their task. Obviously, in many cases, the available lower level behaviors are not appropriate for the higher-level modules to be able to implement the desired global behaviors and thus it is necessary to simultaneously obtain another low-levelbehavior that complements those already present and performs every task not covered by them. By iterating this process as required a tree-like control architecture (Fig. 1 left) is obtained. Higher-level modules modify the operation of lower level modules by acting on their inputs or their outputs through product operations. We have called these processes input / sensor modulation and output / actuator modulation. More formally, being X and Y any two modules:

- X is an ancestor of Y if there is a path from X to Y.
- X is a descendant of Y if there is a path from Y to X.
- X is a direct descendant if there is a path of length 1 from Y to X.
- X will be called a Root node (denoted as R) if it has no ancestors. There can be more than a root node.
- X is an actuator node (A) if its outputs establish values for the actuators.
- X is an actuator modulating node (AM) if its outputs modify (multiplying by a value between 0 and 2) the outputs of its descendant nodes of type A. The modulations propagate through the controller hierarchy until they reach the actuator nodes in such a way that if between R and A there is more than one AM that modulates one output of A, the resulting modulating value will be the product of the individual modulations in the path. Assuming that AM wishes to modulate the values of *n* actuators, its number of outputs must necessarily be *n*number of direct descendants*, as the modulation propagated to each descendant is usually different. When more than one node A provides values for the same actuator, the actuator receives the sum of these values. An AM does not necessarily modulate all the actuators over which the set of nodes acts, just any subset of them.

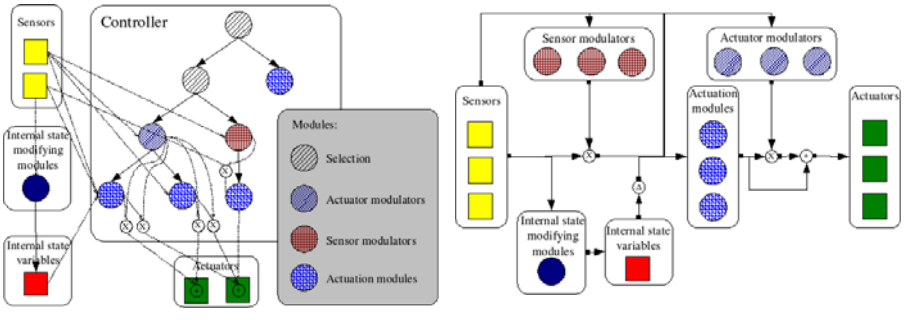


Fig. 1. Left: Example of a controller with all the elements of the architecture. Right: Alternative representation for the architecture (arrows represent blocks of connections)

- X is a sensor modulating node (SM) if its outputs modify (multiplying by a value between 0 and 2) the inputs of its descendant nodes. The modulations propagate through the controller hierarchy until they reach the actuator nodes in such a way that if between R and Y there is more than one SM that modulates one input of Y, the resulting modulating value will be the product of the individual modulations in the path. Assuming that SM wishes to modulate the values of *n* sensors, its number of outputs must necessarily be *n*number of direct descendants*, as the modulation propagated to each descendant is usually different. An SM does not necessarily modulate all the sensors over which the set of nodes act, just any subset of them.
- X is a selector node (S) if its output selects one of its direct descendants as the branch to follow, short-circuiting the others. This is a particular case of an AM that modulates with a value of 1 every actuator of one descendant and with 0 every actuator of the remaining descendants but, if we want this behavior, evolution is simplified by identifying the node as a selector drastically reducing ANN outputs.

- X is a continuous selector node (CS) if the node output modulates one of its descendants with a given value and the other descendants are modulated by its complementary. This node is another particular case of AM.

The use of actuator modulators leads to a continuous range of behaviors for the transitions between those determined by the individual controllers. In addition, sensor modulators permit changing how a module reacts under a given input pattern transforming it to a different one. This way, it is very easy to make changes in that reaction for already learnt input patterns. The construction of the control architecture produces an apparently hierarchical structure, but this is due to the incremental nature of this process. Taking into account the way modulators act, the control architecture is, in fact, distributed as shown in the alternative (functional) representation of Fig.1 right.

3 Collective Supply Example

The present architecture and design approach has been used to obtain the controllers for groups of robots that had to perform different tasks. Here we are going to concentrate on one of these scenarios and use it as a way to demonstrate how the progressive development of complex group behaviors can be achieved through a sequence of simpler steps that lead to the production of the elements that make up the modulating architecture controlling each individual. For the sake of clarity, in this example we are going to assume that all the robots participating in the task use the same controller (although this does not necessarily have to be this way).

We have a scenario where a group of simulated robots must provide humanitarian aid in an area with several villages in conflict that must be supplied. In this scenario there is a *resource level*associated to each village representing the scarcity of goods for basic subsistence. The higher this level, the higher the urgency of supplying the village, and the robots must visit it sooner. In addition, there isa *danger level* associated to the whole area that represents how risky is to visit the villages providing aid. When this danger level rises, the robots must travel close together to support each other in case of attack, that is, as a convoy. If this level is low, each robot can deliver the goods alone without risk. To solve the task, the robots have sensors for the resource level of each village and danger level of the area (simulated as a communication), they are capable of seeing other robots in their surroundings and detectingthe villages' exact position. The global task consists on providing the humanitarian aid as soon as possible assuming the lowest risk for the robot's integrity.

The particular modulation architecture for this example is shown in Fig. 2 and was constructed from the bottom up, by starting with the basic low-level behaviors. These basic behaviors are devoted to the motion behavior of a robot in order to ignore orapproachobjects in the environment, in this case, villages or robots. Above that, it was necessary to implement two modulation levels (see Fig. 2) that will be discussed in the following sections.

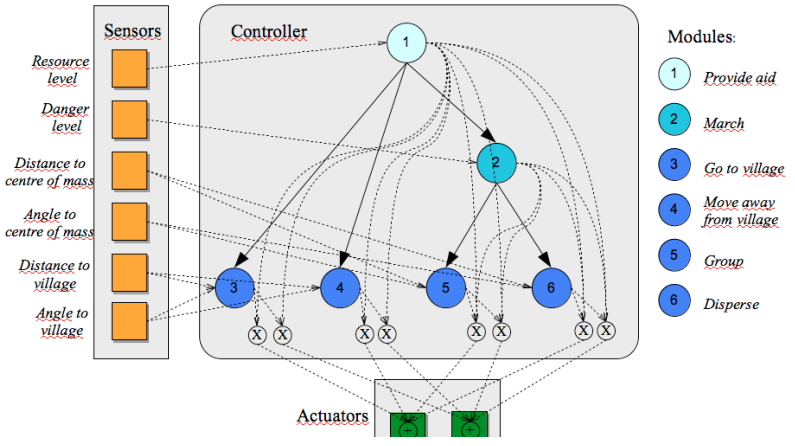


Fig. 2. Controller used in the example

3.1 Low-Level Behaviors

As indicated above, there are two different relevant elements in the environment: robots and villages that need aid. Depending on the danger level of the area, the robots can travel grouped or isolated, and depending on the resource level of each village, the robots can approach it or ignore it. As a consequence, there are four low-level behaviors that can be identified, that is, reaching or escaping from robots or villages. These lower level behaviors could be generalized into two:

- *Reach an object*: the inputs are the distance and the angle to the center of mass of the group of similar objects in the visual field of the robots. The outputs are the linear and angular speeds to reach the objective.
- *Escape from an object*: the inputs and outputs are the same as before but, in this case, the outputs should lead the robot away from the objective.

These controllers are constructed as delay-based ANNs [10] with the same characteristics: 2 inputs, one hidden layer of 4 neurons and 2 outputs. The inclusion of temporal processing ANNs was necessary due to the dynamic characteristics of the experiment, where the robots must modify their direction in real time. Each of the two basic behaviors was obtained by evolution of the ANNs using a canonic genetic algorithm with a population of 240 individuals. During the evolutionary process the fitness is calculated through the addition of particular utility values for each action applied by each robot during 100 steps (100 actions applied) for 10 different initial conditions (initial positions). The utilities have been defined so that approaching or escaping from the objective (depending on the desired behavior for the controller) sums utility value. As an example of the followed procedure, for the approaching behavior:

$$fitness_{reach} = \sum_{i=0}^{n} \left(\left(d_{prev}^{i} - d_{post}^{i}\right) + 10 \cdot e^{\frac{-4 \cdot d_{post}^{i}}{d_{max}^{i}}} \right), \quad (1)$$

where n is the number of steps, $d_{prev}$is the distance to the objective prior to executing the action,d_{post} is the distance to the objective after applying the action and $d_{max}$is the maximum distance to the objective.

In Fig.3 we have represented the evolution of the fitness value of the best individual and the mean fitness value of the population through the generations for these two ANN controllers. As shown in the graph, in about 250 generations, the evolution reaches a high fitness level, which implies that these simple behaviors are successfully achieved by the robots. The fitness value oscillates because it is being calculated as an aggregation of the values in several different runs.

As a conclusion, we can say that the selected low-level controllers are, as expected, easy to evolve and the ANNs that represent them contain a small number of neurons. This implies that the control is simple and accurate.

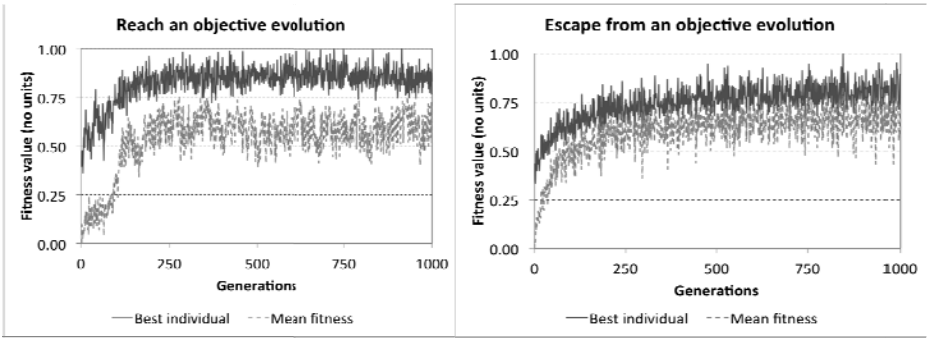


Fig. 3. Evolution of the fitness value of the best individual and the mean fitness of the population through the generations for the controllers involved in approaching (left) and escaping from (right) an objective

3.2 High-Level Behaviors

Once we have evolved the basic low-level behaviors the robots are able to reach or escape from villages and other robots, depending on whether the sensorial inputs correspond to the distance to the nearest village or the center of mass of the neighboring robots (represented in Fig. 2 by nodes 3, 4, 5 and 6). The next step consists in controlling the robot team behavior as a function of the danger level of the area, so that the robots travel isolated or as a group. To achieve this middle-level behavior, we have evolved a continuous selector node(CS) that acts as a modulator controlling the two low-level behaviors of reaching or escaping from other robots(ignoring them) depending on the danger level input value (node 2 in Fig. 2). Thedanger level varies from 0 (no danger) to 10 (maximum risk) and the desired response is that each robot must try to reach the center of mass of its set of neighbors in case of maximum risk and escape from it in case of minimum risk.

This modulator node is represented by a multilayer perceptron ANN with one input, the danger level, one hidden layer with two neurons and one output that selects what proportion of the low level controllers must be used (normalized between 0 and 1). It was obtained using a genetic algorithm with a population of 80 individuals and with just two extreme values for the input, 0 and 10.

Fig. 4. Evolution of the continuous selector module

Fig.4 (left) represents the evolution of the modulator. It can be observed that in about 25 generations the controller reaches almost the highest fitness level value. As an example of the response of thiscontroller, Fig. 4 (right)displays the behavior of the robot team in a simulation. From top to bottom, the danger level increases andthe robots tend to get closer together in a continuous manner.

The second and last level of modulation depends on the resource level of the villages, being the most prioritarymagnitude for this particular task. As a consequence, this module must control the low level behaviors of reaching or escaping from a village and, in addition, it must control the middle-level behavior obtained above that controls the behavior of the team depending on the danger level.

In this case, an actuator modulating (AM) node (module 1 in Fig. 2) implemented using a multilayer perceptron ANN with one input, the resource requirement level, two hidden layers with 3 neurons each and 3 outputs was used. The first output modulates the approaching village controller, the second the escaping from village controller and the third the output of the middle-level CS controller. It was obtained using a genetic algorithm with a population of 130 individuals and with two extreme values for the resource requirement level input: for 10, maximum resource requirement, the modulator should use the output provided by the low-level controller that approaches the villages as aid is urgent and ignore the other two modules, whereas when the resourcerequirement level is 0, the modulator should increase the relevance of the low-level controller that ignores the villages and the danger level.

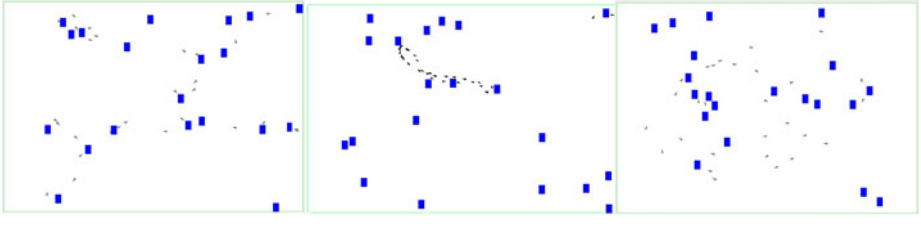


Fig. 5. Behaviors provided by the complete controller for different resource and danger levels

Evolution was successful again and the behaviors are displayed in Fig. 5 as a function of the resource and danger levels. From left to right: resource requirement level of 10 and a danger level of 0, resource requirement and danger levels of 10 andresource requirement and danger levels of 0.

4 Conclusions

This paper provides a modulation based ANN architecture that facilitates the combination of several ANN based controllers into a single behavioral controller for robots. The architecture facilitates the progressive construction of complex controllers, especially when considering evolution. This structured way of doing things permits using simple environments and usually single robot simulations to obtain low level behaviors and then increase complexity by adding controllers that modulate these low level controllers or other controllers below them leading to growing behavior based controllers in an incremental manner.

Acknowledgements

This work was funded by the Xunta de Galicia and Feder through project 09DPI012166PR.

References

1. Kolling, A., Carpin, S.: Multi-robot Surveillance: an Improved Algorithm for the GRAPH-CLEAR Problem. In: Proc. 2008 IEEE ICRA Pasadena, CA, USA, pp. 2360–2365 (2008)
2. Fernandez-Gauna, B., Lopez-Guede, J.M., Zulueta, E., Graña, M.: Learning hose transport control with Q-learning. Neural Network World 20(7), 913–923 (2010)
3. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration Robotics. IEEE Transactions on Robotics 21(3), 376–386 (2005)
4. Folgado, E., Rincón, M., Álvarez, J.R., Mira, J.: A Multi-robot Surveillance System Simulated in Gazebo. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2007. LNCS, vol. 4528, pp. 202–211. Springer, Heidelberg (2007)
5. Waibel, M., Keller, L., Floreano, D.: Genetic Team Composition and Level of Selection in the Evolution of Multi-Agent Systems. IEEE TEC 13(3), 648–660 (2009)
6. José Antonio Martín, H., de Lope, J., Maravall, D.: Learning to coordinate multi-robot competitive systems by stimuli adaptation. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5602, pp. 362–371. Springer, Heidelberg (2009)
7. Cavalcanti, A., Freitas Jr., R., Kretly, L.: Nanorobotics Control Design: A Practical Approach Tutorial, Robotics Today, SME, 4th Quarter, vol.18(4) (2005)
8. Graña, M., Torrealdea, F.J.: Hierarchically structured Systems. European Journal of Operational Research 25, 20–26 (1986)
9. Becerra, J.A., Bellas, F., Santos, J., Duro, R.J.: Complex Behaviours through modulation in Autonomous Robot Control. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 717–723. Springer, Heidelberg (2005)
10. Duro, R.J., Santos, J.: Discrete Time Backpropagation for training Synaptic Delay Based Artificial Neural Networks. IEEE Trans. on Neural Networks 10, 779–789 (1999)

TOPOS 2: Spiking Neural Networks for Bipedal Walking in Humanoid Robots

Pablo González-Nalda and Blanca Cases

Computer Languages and Systems, University of the Basque Country (UPV/EHU)
pablo@si.ehu.es, blancarosa.cases@ehu.es
http://lsi.vc.ehu.es/pablogn/

Abstract. This work analyses the state of the art in the field of Evolutionary Robotics and marks the path we select in the design of evolutionary strategies. The aim of this text is to describe the lines that we are going to follow in the foreseeable future. Our goal is to create through evolution the neural network that couples with a complex humanoid robot body. For us the problems of a non-structured environment and of Evolutionary Robotics need a sub-symbolic conexionist approach based in *Nouvelle AI* that can cope with the coupling among sensorimotor, neural and environment parts. We also describe the tools we choose to accomplish this task.

Keywords: Evolutionary Robotics, Spiking Neural Networks, Artificial Life, sub-symbolic, small-world topologies, CUDA.

1 Introduction

Current computer power brings us the opportunity to search for new paths in Robotics but most of the research is based on the classical Artificial Intelligence paradigm, which relies on a top-down human design. Sometimes the problems are too complex to solve them by hand. The alternative is to build systems with a strong bio-inspired bottom-up structure, to let the evolutionary algorithms to select the appropriate links and relations among the basic parts, taking into account new theories in the field of graphs and networks. In this work we propose a system to exploit the computer power and get not-so-simple behaviours from very low-level primitives with the aim to obtain a truly scalable system.

Evolutionary Robotics [1] has its roots in Artificial Intelligence [2,3], Artificial Life [4], evolutionary strategies [5,6], and in neural networks [7].

2 State of the Art

Our recent work [8,9,10] describes the state of the art in Evolutionary Robotics. ER is a type of Behaviour-based Robotics and hence there are two forms of understanding: the classical approach to robotics and the line defined by Rodney

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part II, LNAI 6679, pp. 479–485, 2011.

Brooks to avoid the problems caused by the use of representations. Ronald Arkin depicts these ideas as a continuum in a spectrum. The Spectrum of Robot Control [11] represents the range of of robot control strategies from a deliberative symbolic, classic AI, high level cognitive approach for a structured environment and a reactive, sub-symbolic, *Nouvelle AI* conexionist based system that can cope and couple with a non-structured environment. The latter is more suitable for a embodied and situated robot controller since it applies the Symbol Grounding Hypothesis, and the former excludes the connection between the world and the system [12] because the designer has to set symbol primitives to *represent* the atomic stimuli.

Embodiment and situatedness are two ideas that we have to take into account if we want to get a robot for a non-structured environment, because the information that the robot needs is closely related to the filtering that the body and sensors physically do. There is a coevolution between the neural system and the shape and function of the sensors. The body as a whole determines and modifies the given response to a stimulus.

Evolutionary Robotics is a young discipline that has followed this approach achieving complex behaviours that emerge from simple interactions among the parts of the system. Nevertheless, there are two problems that have to be addressed in order to obtain more powerful robots: morphogenesis and scalability. First attempts of morphogenesis were a simple and direct expression of the robot parameters and neural weights from the genetic information to the actual controller and the physical robot. This is not a scalable system. Genetic Algorithms cannot find the correct relations among the values that shape the individual. Instead of this, the evolutionary algorithm has to set as an individual the set of parameters of a dynamical system whose final result is the robot. In our opinion, this is the way to develop a body and sensorimotor structure coupled with the topology of the neural network, that can be parametrized and evolved.

In other words, if the environment is complex and it only can be described in a non-linear form, the robot and its evolution has to be equally complex, as a new born child that has to learn to coordinate vision, proprioception and environment. The whole mechanism has to be scalable to allow the growing of the robot capabilities to quantitatively more complex ones.

Some authors have created new models of neural networks, more bio-inspired and with a rich dynamics, time delays and interesting effects on computational capabilities [7,13,14,15]. There has been also a lot of exciting new work on network topologies [16,17,18] that has been applied to Neural Networks but in a theoretical level to study their properties [19,20]. As a clue on how the neuroscience could change with these theories, in [20] we can read:

> *"Our results suggest that mammalian cortical networks, by virtue of being both small-world and topographically organized, seem particularly well-suited to information processing through polychronization."*

It is clear that an experiment in Evolutionary Robotics could help to the progress in these areas.

If this knowledge could be applied to Evolutionary Robotics, within the described framework the so called small-world networks and their algorithms could give us the tools to tackle both problems, morphogenesis and scalability. With these ideas, through simple algorithms the net would grow and connect to form the sensorimotor loop, parametrized with the help of the evolutionary strategies.

The lines that Di Paolo [21] marks as the future of Evolutionary Robotics are on one hand, more and more high-level cognitive behaviours, and on the other hand, experiments as a synthetic biology:

> *"The purpose of work in ER is less centred on trying to obtain more 'cognitively' complex performance —a goal that has not been abandoned— and more on understanding other dimensions of adaptation and the role of different kinds of underlying mechanisms. The design and study of novel integrated systems of this sort may well be one way for evolutionary robotics to contribute useful information back to biology, especially neuroscience, in the proximate future."*

Both objectives could merge and obtain complex cognitive mechanisms based on the last discoveries in neuroscience, with the help of graph theory and new methods of describing, building and connecting the body and the control of the robot.

3 Objectives

After the critical analysis of the situation in Evolutionary Robotics we can describe the main goal that we want to achieve. We want to build mechanisms to get an advance in complex sensorimotor behaviours (as bipedal walking of a humanoid robot) in a completely sub-symbolic conexionist system, that is, the designer deals with neuron definitions, without setting architectures, hierarchies, or at least not directly, but through bio-inspired developmental algorithms. Nevertheless, the current works based on human designed behaviours implement the analytic approaches of the classic AI [22]. These ones are obviously interesting but lack a synthetic framework like the one described. The work we propose is ambitious within the constraints of the theoretical proposal.

For this task we need a big spiking neural network to connect to simulated muscles as a proprioceptive and motor system in a humanoid robot with tens of degrees of freedom. The neural network has to have thousands of neurons because it is easier to extract and set the correct information for each joint, and this topology also has to reflect the somatotopic arrangement (spatial organisation that maintains the body topology within the brain [23]) and to show a small-world connectivity [24].

So this development would give a step further in the fields of morphogenesis, scalability, robotic neuroscience and globally in Evolutionary Robotics. The interaction between the body and the growing network could be the way of emerging a meaningful structure of the neural network in the previously described framework and consequently at least a partial solution to the problems that usually arise when building new robots using evolution.

4 Tools

The first thing we need is a simulation of a humanoid robot. The need of a simulation is because it is much easier and cheaper to apply evolutionary strategies in simulation due to the available computational power and the lack of damages in the robot. After a search we choose SimSpark simulator [25], the one used in RoboCup 3D Soccer Simulation League. The reasons are that is free software, it uses the Open Dynamics Engine (ODE) for simulating physics as velocity, inertia and friction, rigid body dynamics and collisions. It has a model of a Nao-like robot with a lot of sensors and 22 degrees of freedom (see fig. 1).

Fig. 1. A Nao-like humanoid robot in the SimSpark simulation

A software that helps to use SimSpark is the `libbats` library [26], which also it is free software (GPL). This library deals with some common operations that we do not need to change from the usual way and let us to read and set joint angles easily.

We also need a fast simulation of a spiking neural network. We select the Izhikevich model [14]. The reasons to pick out this one are explained in [27]. In short, this model is one of the fastest and also it has most of the necessary characteristics. The formulas that rule the neuron dynamics are the following (eq. 1).

$$
\begin{aligned}
\dot{v}_i &= 0.04v_i^2 + 5v_i + 140 - u_i + I, \\
\dot{u}_i &= a(bv_i - u_i), \\
\text{if } v_i &\geq 30\ mV \\
v_i &\leftarrow c, \\
u_i &\leftarrow u_i + d.
\end{aligned} \tag{1}
$$

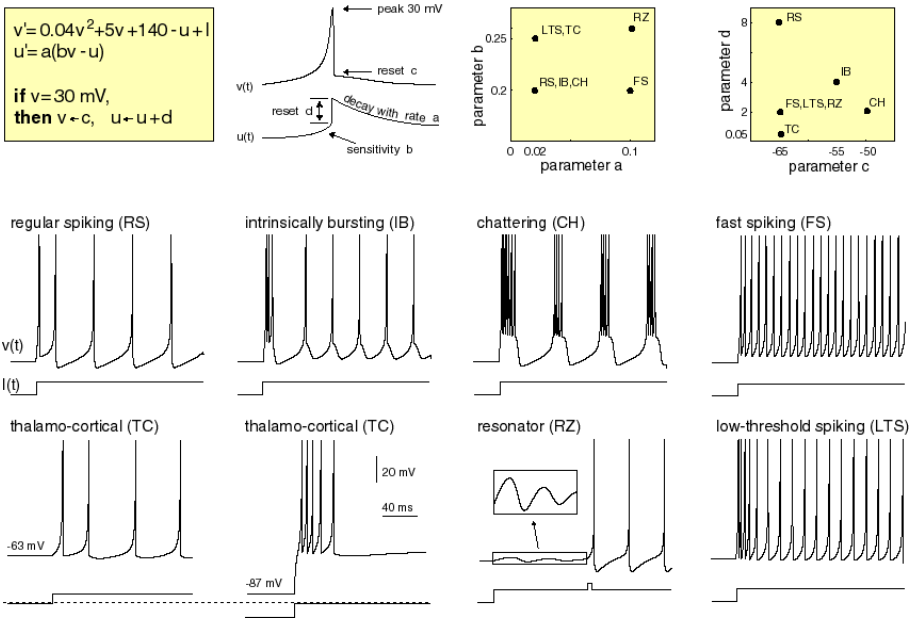


Fig. 2. Some of the neuron dynamics in the Izhikevich model, from [14]

The robot controllers for the SimSpark are written in `C++`, so we can run them at high speed, but the neural networks are fully "parallelisable", that is, they are matrix calculus that can be done in parallel. Nowadays, we have powerful rendering graphical processor units (GPU) that are designed to calculate at each processing core the colour, brightness and shadows for each pixel in a 3D scene. There are programs that allow to send non-graphical computing to GPU to perform general-purpose computing. In this case the hardware and software pack that we have chosen is CUDA [28]. It is a mature technology and it is demonstrated that can be 26 times faster than a common CPU software when simulating a neural network based on the Izhikevich model [29].

The design of the *fitness functions* is always a problem in Evolutionary Robotics. We have to design an experiment in which a function has to tell how good is a robot (one of the possible solutions of a problem subject to evolution)

in order to select or discard the parameters that define the robot. In this case it is easy to design the experiment thanks to the problem that we want to work, the bipedal locomotion. The fitness value can be calculated as the covered distance in a fixed time. It could be also added a value related to the time that the robot has maintained the standing position while walking.

5 Conclusions

In this work we have set the basis for a search of new techniques in Robotics. We continue our previous in Evolutionary Robotics, remarking the bio-inspired bottom-up structure, and adding new works on graphs to permit evolutionary strategies to build the neural controller of the robots that fits with a complex humanoid robot body.

References

1. Harvey, I.: The artificial evolution of adaptive behaviour. Ph.D. thesis, University of Sussex (1995)
2. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall, Englewood Cliffs (1994)
3. Brooks, R.: Intelligence without representation. Artificial Intelligence 47, 139–159 (1991)
4. Langton, C.G. (ed.): Proc. First Intl. Conf. on Artificial Life. Addison-Wesley, Redwood City (1989)
5. Holland, J.: Adaptation in Natural and Artificial Systems. U. Michigan Press, Ann Arbor (1975)
6. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Weley, Reading (1989)
7. Maass, W., Bishop, C.M. (eds.): Pulsed Neural Networks. MIT Press, Cambridge (1999)
8. González-Nalda, P., Cases, B.: Topos: Spiking neural networks for temporal pattern recognition in complex real sounds. Neurocomputing 71(4-6), 721–732 (2008), http://dx.doi.org/10.1016/j.neucom.2007.07.032
9. González-Nalda, P., Cases, B.: Topos: generalized Braitenberg vehicles that recognize complex real sounds as landmarks. In: Alife X, pp. 213–219. MIT Press, Cambridge (2006)
10. González-Nalda, P.: Navegación mediante Evolución de Redes Neuronales Recurrentes y Dinámicas. Ph.D. thesis, University of the Basque Country (2008), http://lsi.vc.ehu.es/pablogn/investig/ficheros/PGN-Tesis.pdf
11. Arkin, R.: Behavior-Based Robotics. MIT Press, Cambridge (1998)
12. Brooks, R.: Elephants don't play chess. Robotics and Autonomous Systems 6, 3–15 (1990)
13. Maass, W.: Networks of spiking neurons: the third generation of neural network models. Neural Networks 10, 1659–1671 (1997)
14. Izhikevich, E.M.: Simple model of spiking neurons. IEEE Trans Neural Netw. 14(6), 1569–1572 (2003)
15. Izhikevich, E.M.: Polychronization: Computation with spikes. Neural Comput. 18(2), 245–282 (2006)

16. Watts, D.J., Strogatz, S.H.: Nature 393, 440 (1998)
17. Barabási, A.L., Albert, R.: Science 286, 509 (1999)
18. Amaral, L.A.N., Ottino, J.M.: Eur. Phys. J. B., 147–162 (2004)
19. Kwok, H.F., Jurica, P., Raffone, A., van Leeuwen, C.: Robust emergence of small-world structure in networks of spiking neurons. Cogn. Neurodyn. 1, 39–51 (2007)
20. Vertes, P.E., Duke, T.: Effect of network topology on neuronal encoding based on spatiotemporal patterns of spikes. HFSP J. 4(3-4), 153–163 (2010)
21. Di Paolo, E.A.: Evolutionary robotics (S. Nolfi and D. Floreano). In: Nolfi, S., Floreano, D. (eds.) Connection Science, cap. Review, vol. 14, MIT Press, Cambridge (2002)
22. Picado, H., Gestal, M., Lau, N., Tom, A.M., Reis, L.P.: Grid Automatic Generation of Biped Walk Behaviour Using Genetic Algorithms. In: IWANN 2009, pp. 805–812. Springer, Heidelberg (2009)
23. Manni, Petrosini, E.L.: A century of cerebellar somatotopy: a debated representation. Nature Reviews Neuroscience 5, 241–249 (2009)
24. Smith-Bassett, D., Bullmore, E.: Small-world brain networks. Neuroscientist 12(6), 512–523 (2006)
25. Obst, O., Rollmann, M.: Spark - A generic simulator for physical multi-agent simulations. Computer Systems Science and Engineering, 20(5) (2005), http://sourceforge.net/projects/simspark/
26. The Little Green BATS, http://www.littlegreenbats.nl: Libbats Library: https://launchpad.net/littlegreenbats
27. Izhikevich, E.M.: Which model to use for cortical spiking neurons? IEEE Trans. Neural Netw. 15(5), 1063–1070 (2004)
28. NVIDIA Corporation. CUDA Programming Guide for CUDA Toolkit 3.2 (2010)
29. Nageswaran, J.M., Dutt, N., Krichmar1, J.L., Nicolau, A., Veidenbaum, A.V.: A Configurable Simulation Environment for the Efficient Simulation of Large-Scale Spiking Neural Networks on Graphics Processors. Neural Networks 22, 791–800 (2009)

Author Index