Emilio Corchado Václav Snášel
Ajith Abraham Michał Woźniak
Manuel Graña Sung-Bae Cho (Eds.)

LNAI 7209

Hybrid Artificial Intelligent Systems

7th International Conference, HAIS 2012
Salamanca, Spain, March 2012
Proceedings, Part II

2 Part II

HAIS 2012

Springer

Lecture Notes in Artificial Intelligence 7209

Subseries of Lecture Notes in Computer Science

Emilio Corchado Václav Snášel
Ajith Abraham Michał Woźniak
Manuel Graña Sung-Bae Cho (Eds.)

Hybrid Artificial Intelligent Systems

7th International Conference, HAIS 2012
Salamanca, Spain, March 28-30, 2012
Proceedings, Part II

Springer

Volume Editors

Emilio Corchado
University of Salamanca, Spain
E-mail: escorchado@usal.es

Václav Snášel
VŠB-TU Ostrava, Czech Republic
E-mail: vaclav.snasel@vsb.cz

Ajith Abraham
Machine Intelligence Research Labs, Washington, DC, USA
E-mail: ajith.abraham@ieee.org

Michał Woźniak
Wroclaw University of Technology, Poland
E-mail: michal.wozniak@pwr.wroc.pl

Manuel Graña
University of the Basque Country, San Sebastian, Spain
E-mail: ccpgrrom@si.ehu.es

Sung-Bae Cho
Yonsei University, Seoul, Korea
E-mail: sbcho@cs.yonsei.ac.kr

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-28930-9 e-ISBN 978-3-642-28931-6
DOI 10.1007/978-3-642-28931-6

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011927871

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

This volume of Lecture Notes in Artificial Intelligence (LNAI) includes accepted papers presented at the 7th HAIS held in the beautiful and historic city of Salamanca, Spain, in March 2012.

The International Conference on Hybrid Artificial Intelligence Systems (HAIS 2012) has become a unique, established and broad interdisciplinary forum for researchers and practitioners who are involved in developing and applying symbolic and sub-symbolic techniques aimed at the construction of highly robust and reliable problem-solving techniques to present the most relevant achievements in this field.

Hybridization of intelligent techniques, coming from different computational intelligence areas, has become popular because of the growing awareness that such combinations frequently perform better than the individual techniques such as neurocomputing, fuzzy systems, rough sets, evolutionary algorithms, agents and multiagent Systems, among others.

Practical experience has indicated that hybrid intelligence techniques might be helpful to solve some of the challenging real-world problems. In a hybrid intelligence system, a synergistic combination of multiple techniques is used to build an efficient solution to deal with a particular problem. This is, thus, the setting of HAIS conference series, and its increasing success is proof of the vitality of this exciting field.

HAIS 2012 received 293 technical submissions. After a rigorous peer-review process, the International Program Committee selected 118 papers, which are published in these conference proceedings. In this edition a special emphasis was put on the organization of special sessions and workshops. Eight special sessions and one workshop, containing 67 papers in total, were organized on the following topics:

Special Sessions:

- Systems, Man, and Cybernetics by HAIS
- Methods of Classifier Fusion
- HAIS for Computer Security
- Data Mining: Data Preparation and Analysis
- Hybrid Artificial Intelligence Systems in Management of Production Systems
- Hybrid Artificial Intelligent Systems for Ordinal Regression
- Hybrid Metaheuristics for Combinatorial Optimization and Modelling Complex Systems
- Hybrid Computational Intelligence and Lattice Computing for Image and Signal Processing

Workshops:

- Nonstationary Models of Pattern Recognition and Classifier Combinations

The selection of papers was extremely rigorous in order to maintain the high quality of the conference, and we would like to thank the Program Committee for their hard work in the reviewing process. This process is very important to the creation of a conference of high standard and the HAIS conference would not exist without their help.

The large number of submissions is certainly not only testimony to the vitality and attractiveness of the field but an indicator of the interest in the HAIS conferences themselves.

HAIS 2012 enjoyed outstanding keynote speeches by distinguished guest speakers: Tom Heskes, Radboud Universiteit Nijmegen (The Netherlands) and Xindong Wu, University of Vermont (USA).

HAIS 2012 teamed up with the *Neurocomputing* (Elsevier) and the *Applied Soft Computing* (Elsevier) journals for special issues and fast-track publication including selected papers from the conference.

Particular thanks also go to the conference main sponsors, IEEE-Sección España, IEEE Systems, Man and Cybernetics -Capítulo Español, AEPIA, World Federation of Soft Computing, MIR Labs, IT4Innovation Centre of Excellence, The International Federation for Computational Logic, Ministerio de Economía y Competitividad, Junta de Castilla y León, Ayuntamiento de Salamanca, University of Salamanca, who jointly contributed in an active and constructive manner to the success of this initiative. We also want to extend our warm gratitude to all the Special Session and Workshop Chairs for their continuing support of the HAIS series of conferences.

We would like to thank Alfred Hofmann and Anna Kramer from Springer for their help and collaboration during this demanding publication project.

March 2012

Emilio Corchado
Václav Snášel
Ajith Abraham
Michał Woźniak
Manuel Graña
Sung-Bae Cho

Organization

Honorary Chairs

Alfonso Fernández Mañueco	Mayor of Salamanca
Antonio Bahamonde	President of the Spanish Association for Artificial Intelligence (AEPIA)
Pilar Molina	Chair IEEE Spanish Section
Hojjat Adeli	The Ohio State University, USA
Manuel Castro	Past Chair IEEE Spanish Section

General Chair

Emilio Corchado	University of Salamanca, Spain

International Advisory Committee

Ajith Abraham	Machine Intelligence Research Labs, Europe
Antonio Bahamonde	President of the Spanish Association for Artificial Intelligence, AEPIA
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	King's College London, UK
Ali A. Ghorbani	UNB, Canada
Mark A. Girolami	University of Glasgow, UK
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, The Netherlands
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

Marios Polycarpou	University of Cyprus, Cyprus
Witold Pedrycz	University of Alberta, Canada
Václav Snášel	VSB-Technical University of Ostrava, Czech Republic
Xin Yao	University of Birmingham, UK
Hujun Yin	University of Manchester, UK
Michał Woźniak	Wroclaw University of Technology, Poland
Aditya Ghose	University of Wollongong, Australia
Ashraf Saad	Armstrong Atlantic State University, USA
Bernadetta Kwintiana	Universität Stuttgart, Germany
Fanny Klett	German Workforce Advanced Distributed Learning Partnership Laboratory, Germany
Ivan Zelinka	VSB-Technical University of Ostrava, Czech Republic

Industrial Advisory Committee

Rajkumar Roy	The EPSRC Centre for Innovative Manufacturing in Through-life Engineering Services, UK
Amy Neustein	Linguistic Technology Systems, USA
JaydipSen	Innovation Lab, Tata Consultancy Services Ltd., India

Program Committee

Emilio Corchado	University of Salamanca, Spain (Co-chair)
Václav Snášel	VSB-Technical University of Ostrava, Czech Republic (Co-chair)
Ajith Abraham	Machine Intelligence Research Labs, Europe (Co-chair)
Michał Woźniak	Wroclaw University of Technology, Poland (Co-chair)
Manuel Grana	University of the Basque Country/EHU, Spain (Co-chair)
Sung-Bae Cho	Yonsei University, Korea (Co-chair)
Abdel-Badeeh M. Salem	Ain Shams University, Egypt
Aboul Ella Hassanien	Cairo University, Egypt
Adolfo Rodríguez	University of León, Spain
Alberto Fernández	Universidad Rey Juan Carlos, Spain
Alberto Ochoa	Juarez City University, Mexico
Aldo Franco Dragoni	Università Politecnica delle Marche, Italy
Alfredo Cuzzocrea	ICAR-CNR and University of Calabria, Italy
Alicia Troncoso	Pablo de Olavide University, Spain
Álvaro Herrero	University of Burgos, Spain
Amelia Zafra	University of Córdoba, Spain

Ana M. Bernardos	Universidad Politécnica de Madrid, Spain
Ana María Madureira	Polytechnic University of Porto, Portugal
Anca Gog	University of Babes-Bolyai, Romania
André de Carvalho	University of São Paulo, Brazil
Andreea Vescan	University of Babes-Bolyai, Romania
Andrés Ortiz	University of Málaga, Spain
Ángel Arroyo	University of Burgos, Spain
Angelos Amanatiadis	Democritus University of Thrace, Greece
Anna Burduk	Wroclaw University of Technology, Poland
Antonio Bahamonde	University of Oviedo, Spain
António Dourado	University of Coimbra, Portugal
Arkadiusz Kowalski	Wroclaw University of Technology, Poland
Arturo de la Escalera	University Carlos III de Madrid, Spain
Arturo Hernández-Aguirre	CIMAT, Mexico
Barna Iantovics	PetruMaior University of Tg. Mures, Romania
Belén Vaquerizo	University of Burgos, Spain
Bernardete Ribeiro	University of Coimbra, Portugal
Bingyang Zhao	Tsinghua University, China
Blanca Cases Gutierrez	University of the Basque Country/EHU, Spain
Bogdan Trawinski	Wroclaw University of Technology, Poland
Borja Fernandez-Gauna	University of the Basque Country/EHU, Spain
Bożena Skolud	Silesian University of Technology, Poland
Bruno Baruque	University of Burgos, Spain
Camelia Chira	University of Babes-Bolyai, Romania
Camelia Pintea	North University of Baia-Mare, Romania
Carlos Carrascosa	Universidad Politécnica de Valencia, Spain
Carlos D. Barranco	Pablo de Olavide University, Spain
Carlos G. Puntonet	Universidad de Granada, Spain
Carlos Pereira	University of Coimbra, Portugal
Carmen Hernández	University of the Basque Country/EHU, Spain
Carmen Vidaurre	Berlin Institute of Technology, Germany
César Hervás	University of Córdoba, Spain
Cezary Grabowik	Silesian University of Technology, Poland
Constantin Zopounidis	Technical University of Crete, Greece
Cristóbal José Carmona	University of Jaén, Spain
Damian Krenczyk	Silesian University of Technology, Poland
Daniel Mateos-García	University of Seville, Spain
Dante I. Tapia	University of Salamanca, Spain
Dario Landa-Silva	University of Nottingham, UK
Darya Chyzhyk	University of the Basque Country/EHU, Spain
David Iclanzan	Sapientia Hungarian University of Transylvania, Romania
Diego Pablo Ruiz	University of Granada, Spain
Diego Salas-Gonzalez	University of Granada, Spain
Dimitris Mourtzis	University of Patras, Greece
Dominik Slezak	University of Regina, Canada

Donald Davendra	VSB TU Ostrava, Czech Republic
Dragan Simic	University of Novi Sad, Serbia
Dragos Horvath	Université de Strassbourg, France
Eiji Uchino	Yamaguchi University, Japan
Elías Fernández-Combarro	University of Oviedo, Spain
Emilio Corchado	University of Salamanca, Spain
Estefania Argente	University of Valencia, Spain
Eva Lucrecia Gibaja	University of Córdoba, Spain
Fabricio Olivetti de França	University of Campinas, Brazil
Federico Divina	Pablo de Olavide University, Spain
Feifei Song	Peking University, China
Fermín Segovia	University of Granada, Spain
Fernando De La Prieta	University of Salamanca, Spain
Fidel Aznar	University of Alicante, Spain
Florentino Fdez-Riverola	University of Vigo, Spain
Francisco Bellas	University of Coruña, Spain
Francisco Cuevas	CIO, Mexico
Francisco Fernández-Navarro	University of Córdoba, Spain
Francisco Herrera	University of Granada, Spain
Francisco Martínez	University of Córdoba, Spain
Francisco Martínez-Álvarez	Pablo de Olavide University, Spain
Frank Klawonn	Ostfalia University of Applied Sciences, Germany
George Dounias	University of the Aegean, Greece
George Papakostas	Democritus University of Thrace, Greece
Gerardo M. Méndez	Instituto Tecnológico de Nuevo León, Mexico
Gerhard Ritter	University of Florida, USA
Giancarlo Mauri	University of Milan-Bicocca, Italy
Giorgio Fumera	University of Cagliari, Italy
Gloria Cerasela Crisan	Vasile Alecsandri University of Bacau, Romania
Gonzalo A. Aranda-Corral	University of Huelva, Spain
Guiomar Corral	Ramon Llull University, Spain
Guoyin Wang	Chongqing University of Posts and Telecommunications, China
Han Pingchou	Peking University, China
Henrietta Toman	University of Debrecen, Hungary
Honghai Liu	University of Portsmouth, UK
Huiyu Huiyu Zhou	Queen's University Belfast, UK
Ignacio Turias	University of Cadiz, Spain
Indre Zliobaite	Bournemouth University, UK
Inés Galván	University Carlos III de Madrid, Spain
Ingo Keck	University of Regensburg, Germany
Ioannis Hatzilygeroudis	University of Patras, Greece
Irene Díaz	University of Oviedo, Spain
Isabel Barbancho	University of Málaga, Spain
Isabel Nepomuceno	University of Seville, Spain

Ivan Zelinka	Tomas Bata University, Czech Republic
IvicaVeza	University of Split, Croatia
Jacino Mata	University of Huelva, Spain
Jaume Bacardit	University of Nottingham, UK
Javier Bajo	Universidad Pontificia de Salamanca, Spain
Javier de Lope	Universidad Politécnica de Madrid, Spain
Javier R. Pérez	Universidad de Granada, Spain
Javier Sedano	University of Burgos, Spain
Jerzy Grzymala-Busse	University of Kansas, USA
Jerzy Sas	Wroclaw University of Technology, Poland
Jerzy Stefanowski	Poznan University of Technology, Poland
Jesús Alcala-Fernandez	University of Granada, Spain
Joaquín Derrac	University of Granada, Spain
Jorge Díez	University of Oviedo, Spain
Jorge García	University of Seville, Spain
José Dorronsoro	Universidad Autónoma de Madrid, Spain
José García	University of Alicante, Spain
José L. Álvarez	Universidad de Huelva, Spain
Jose Luis Calvo	University of Coruña, Spain
José Luis Martínez	Universidad de Castilla-La Mancha, Spain
José Luis Verdegay	University of Granada, Spain
José M. Armingol	University Carlos III de Madrid, Spain
José M. Molina	University of Seville, Spain
José Manuel López	University of the Basque Country/EHU, Spain
José R. Villar	University of Oviedo, Spain
José Ramón Cano	University of Jaén, Spain
Jose Ranilla	University of Oviedo, Spain
José Riquelme	University of Seville, Spain
Jovita Nenortaite	Kaunas University of Technology, Lithuania
Juan Álvaro Muñoz	University of Almería, Spain
Juan F. De Paz Santana	University of Salamanca, Spain
Juan Humberto Sossa	CIC-IPN, Mexico
Juan José Flores	University of Michoacana, Mexico
Juan M. Corchado	University of Salamanca, Spain
Juan Manuel Gorriz	University of Granada, Spain
Juan Pavón	Universidad Complutense de Madrid, Spain
Julián Luengo	University of Granada, Spain
Julio César Ponce	Universidad Autónoma de Aguascalientes, Mexico
Kamil Krot	Wroclaw University of Technology, Poland
Karmele López de Ipina	University of the Basque Country/EHU, Spain
Katya Rodríguez-Vázquez	Universidad Nacional Autónoma de México, Mexico
Keshav Dahal	University of Bradford, UK
Kevin Knuth	University at Albany, USA
Khaled Ragab	King Faisal University, Saudi Arabia

Konrad Jackowski	Wroclaw University of Technology, Poland
Krzysztof Kalinowski	Silesian University of Technology, Poland
Lars Graening	Honda Research Institute Europe, Germany
Lauro Snidaro	University of Udine, Italy
Lenka Lhotská	Czech Technical University in Prague, Czech Republic
Leocadio González	University of Almería, Spain
Leticia Curiel	University of Burgos, Spain
Li Cheng	University of North Carolina, USA
Lina Petrakieva	Glasgow Caledonian University, UK
Lourdes Sáiz	University of Burgos, Spain
Luis Alonso	University of Salamanca, Spain
Luis Búrdalo	Universitat Politècnica de València, Spain
Maciej Grzenda	Warsaw University of Technology, Poland
Maite García-Sebastián	Fundación CITA-Alzheimer, Spain
Marcilio de Souto	Universidade Federal do Rio Grande do Norte, Brazil
Marcin Zmyślony	Wroclaw University of Technology, Poland
Marco Mora	Universidad Católica del Maule, Chile
María del Mar Martínez	University of Seville, Spain
María Dolores Torres	Universidad Autónoma de Aguascalientes, Mexico
María Guijarro	Universidad Complutense de Madrid, Spain
María José del Jesús	University of Jaén, Spain
María Sierra	University of Oviedo, Spain
Mario Köppen	Kyushu Institute of Technology, Japan
Marta Arias	Universidad Politécnica de Cataluña, Spain
Martí Navarro	Universidad Politécnica de Valencia, Spain
Matjaz Gams	Jozef Stefan Institute Ljubljana, Slovenia
Michał Kuliberda	Wroclaw University of Technology, Poland
Mieczysław Jagodziński	Silesian University of Technology, Poland
Miguel A. Patricio	University Carlos III de Madrid, Spain
Miguel Ángel Veganzones	University of the Basque Country/EHU, Spain
Mohammed Chadli	UPJV, France
Neveen Ghali	Al-Azhar University, Egypt
Nicola Di Mauro	University of Bari Aldo Moro, Italy
Nikos Thomaidis	University of the Aegean, Greece
Nima Hatami	University of Cagliari, Italy
Norberto Díaz	Pablo de Olavide University, Spain
Óscar Ibañez	European Centre for Soft Computing, Spain
Otoniel López	Miguel Hernandez University, Spain
Ozgur Koray Sahingoz	Turkish Air Force Academy, Turkey
Pablo González	University of the Basque Country/EHU, Spain
Paola Mello	University of Bologna, Italy
Paula Castro	University of Coruña, Spain
Pedro Antonio Gutiérrez	University of Córdoba, Spain

Peter Rockett	The University of Sheffield, UK
Peter Sussner	University of Campinas, Brazil
Petrica Pop	North University of Baia-Mare, Romania
Petro Gopych	Universal Power Systems USA, Ukraine
Przemysław Kazienko	Wroclaw University of Technology, Poland
Rafael Alcalá	University of Granada, Spain
Rafael Corchuelo	University of Seville, Spain
Ramón Moreno	University of the Basque Country/EHU, Spain
Ramón Rizo	University of Alicante, Spain
Ricardo del Olmo	University of Burgos, Spain
Richard Duro	University of Coruña, Spain
Richard Freeman	Capgemini, Spain
Robert Burduk	Wroclaw University of Technology, Poland
Roberto Uribeetxeberria	Mondragon University, Spain
Rodica I. Lung	University of Babes-Bolyai, Romania
Rodolfo Zunino	University of Genoa, Italy
Roman Senkerik	Tomas Bata University in Zlin, Czech Republic
Ronald Yager	Iona College, USA
Roque Marin	University of Murcia, Spain
Rubén Fuentes-Fernández	Universidad Complutense de Madrid, Spain
Salvador García	University of Jaén, Spain
Sean Holden	University of Cambridge, UK
Sebastián Ventura	University of Córdoba, Spain
Shanmugasundaram Hariharan	Anna University, India
Soo-Young Lee	Brain Science Research Center, Korea
Stella Heras	Universidad Politécnica de Valencia, Spain
Talbi El-Ghazali	University of Lille, France
Teresa Ludermir	Federal University of Pernambuco, Brazil
Theodore Pachidis	Technological Educational Institution of Kavala, Greece
Tom Heskes	Radboud University Nijmegen, The Netherlands
Tomasz Kajdanowicz	Wroclaw University of Technology, Poland
Ulf Johansson	University of Borås, Sweden
Urko Zurutuza	Mondragon University, Spain
Urszula Markowska-Kaczmar	Wroclaw University of Technology, Poland
Urszula Stanczyk	Silesian University of Technology, Poland
Vasile Palade	Oxford University, USA
Vassilis Kaburlasos	Technological Educational Institution of Kavala, Greece
Vicente Julián	Universidad Politécnica de Valencia, Spain
Waldemar Malopolski	Cracow University of Technology, Poland
Wei-Chiang Samuelson Hong	Oriental Institute of Technology, Taiwan
Wei Yang Dai	Fudan University, China
Wieslaw Chmielnicki	Jagiellonian University, Poland
Yannis Marinakis	Technical University of Crete, Greece

Ying Tan	Peking University, China
Yusuke Nojima	Osaka Prefecture University, Japan
Zuzana Oplatkova	Tomas Bata University in Zlin, Czech Republic

Special Sessions

Systems, Man, and Cybernetics by HAIS

Emilio Corchado	University of Salamanca, Spain
Manuel Graña	University of the Basque Country/EHU, Spain
Richard Duro	University of Coruña, Spain
Juan M. Corchado	University of Salamanca, Spain
Vicent Botti	Polytechnical University of Valencia, Spain
Ramón Rizo	University of Alicante, Spain
Juan Pavón	University Complutense of Madrid, Spain
José Manuel Molina	University Carlos III of Madrid, Spain
Francisco Herrera	University of Granada, Spain
César Hervás	University of Córdoba, Spain
Sebastian Ventura	University of Córdoba, Spain
Álvaro Herrero	University of Burgos, Spain
Bruno Baruque	University of Burgos, Spain
Javier Sedano	University of Burgos, Spain
Sara Rodríguez	University of Salamanca, Spain
Lourdes Sáiz Barcena	University of Burgos, Spain
Ana Gil	University of Salamanca, Spain
Héctor Quintián	University of Salamanca, Spain
José Luis Calvo Rolle	University of Coruña, Spain
María Dolores Muñoz	University of Salamanca, Spain
Ángel Arroyo	University of Burgos, Spain

Methods of Classifier Fusion

Emilio Corchado	University of Salamanca, Spain
Bruno Baruque	University of Burgos, Spain
Michał Woźniak	Wroclaw University of Technology, Poland
Václav Snášel	VSB-Technical University of Ostrava, Czech Republic
Bogdan Trawinski	Wroclaw University of Technology, Poland
Giorgio Fumera	University of Cagliari, Italy
Konrad Jackowski	Wroclaw University of Technology, Poland
Konstantinos Sirlantzis	University of Kent, UK
Robert Burduk	Wroclaw University of Technology, Poland
Urszula Stanczyk	Silesian University of Technology, Poland
Przemysław Kazienko	Wroclaw University of Technology, Poland
Jerzy Stefanowski	Poznan University of Technology, Poland
Julián Luengo	University of Burgos, Spain

Balint Antal	University of Debrecen, Hungary
Hadju Andras	University of Debrecen, Hungary
Tom Heskes	Radboud University Nijmegen, The Netherlands
Leticia Curiel	University of Burgos, Spain

HAIS for Computer Security (HAISFCS)

Emilio Corchado	University of Salamanca, Spain
Álvaro Herrero	University of Burgos, Spain
Ángel Arroyo Puente	University of Burgos, Spain
Carlos Laorden	University of Deusto, Spain
Ignacio Arenaza	Mondragon University, Spain
Igor Santos Grueiro	University of Deusto, Spain
Manuel Jacinto Martínez	Ibermática, Spain
Valentina Casola	Università degli Studi di Napoli Federico II, Italy
Juan Álvaro Muñoz Naranjo	University of Almería, Spain
Amparo Fúster-Sabater	Institute of Applied Physics, Spain
Petro Gopych	Universal Power Systems USA, Ukraine
Raquel Redondo	University of Burgos, Spain
Urko Zurutuza	Mondragon University, Spain
Xiuzhen Chen	Shanghai Jiaotong University, China
Wenjian Luo	University of Science and Technology of China, China
Héctor Alaiz Moretón	University of León, Spain
Juan Jesús Barbarán Sánchez	University of Granada, Spain
Luis Hernández Encinas	Consejo Superior de Investigaciones Científicas, CSIC, Spain
Juan Tapiador	University of York, UK
Belén Vaquerizo	University of Burgos, Spain
Bernardete Ribeiro	University of Coimbra, Portugal
Joaquín García-Alfaro	Carleton University, Canada
Juan Manuel González Nieto	Queensland University of Technology, Australia
Ricardo Contreras Arriagada	Universidad de Concepción, Chile
Wei Wang	Norwegian University of Science and Technology, Norway
Paul Axayacatl Frausto	Mediscs, France
SeemaVerma	Banasthali University, India

Data Mining: Data Preparation and Analysis

Salvador García	University of Jaén, Spain
Julián Luengo	University of Burgos, Spain
Francisco Herrera	University of Granada, Spain

Antonio Rivera	University of Jaén, Spain
Cristóbal J. Carmona	University of Jaén, Spain
Isaac Triguero	University of Granada, Spain
José A. Sáez	University of Granada, Spain
Mikel Galar	Public University of Navarra, Spain
Victoria López	University of Granada, Spain
Alberto Fernández	University of Granada, Spain
Jose Antonio Sanz	Public University of Navarra, Spain
Ana M. Martínez	Universidad de Castilla-La Mancha, Spain
Habiba Drias	USTHB, Algeria
Jesús Alcalá-Fdez	University of Granada, Spain
Joaquín Derrac Rus	University of Granada, Spain
Jose R. Villar	University of Oviedo, Spain
Sergio Esparcia	Universidad Politécnica de Valencia, Spain
Stefanos Ougiaroglou	University of Macedonia, Greece
José García Moreno	University of Granada, Spain
Nenad Tomasev	Jozef Stefan Institute, Stovenia
Rafael del Hoyo	Technological Institute of Aragón, Spain
Krystyna Napierala	Poznan University of Technology, Poland
Jose Ramón Cano	University of Jaén, Spain
Aida Gema de Haro	University of Córdoba, Spain
Ana Palacios	University of Oviedo, Spain
Antonio Jesus Rivera	University of Jaén, Spain
Kim Hee-Cheol	Inje University, Korea
Miguel García Torres	Pablo de Olavide University, Spain
Núria Macià	Universitat Ramon Llull, Spain
Rubén Jaramillo	LAPEM-CIATEC, Spain
Olgierd Unold	Wroclaw University of Technology, Poland
Pablo Bermejo	Universidad de Castilla-La Mancha, Spain
Philippe Fournier-Viger	University of Moncton, Canada
Yong Shi	Kennesaw State University, USA

Hybrid Artificial Intelligence Systems in Management of Production Systems

Edward Chlebus	Wroclaw University of Technology, Poland
Milan Gregor	University of Žilina, Slovak Republic
Ulrich Günther	Dresden University of Technology, Germany
Adam Hamrol	Poznan University of Technology, Poland
Bożena Skolud	Wroclaw University of Technology, Poland
Anna Burduk	Wroclaw University of Technology, Poland
Arkadiusz Kowalski	Wroclaw University of Technology, Poland
Cezary Grabowik	Wroclaw University of Technology, Poland
Kamil Krot	Wroclaw University of Technology, Poland
Krzysztof Kalinowski	Wroclaw University of Technology, Poland
Miczyslaw Jagodzinski	Wroclaw University of Technology, Poland

Tomasz Chlebus	Wroclaw University of Technology, Poland
Michał Kuliberda	Wroclaw University of Technology, Poland
Damian Krenczyk	Wroclaw University of Technology, Poland
DimitrisMourtzis	University of Patras, Greece

Hybrid Artificial Intelligent Systems for Ordinal Regression

César Hervás	University of Córdoba, Spain
Pedro Antonio Gutiérrez-Peña	University of Córdoba, Spain
Jaime S. Cardoso	University of Porto, Portugal
Francisco Fernández-Navarro	University of Córdoba, Spain
Francisco Martínez-Estudillo	University of Córdoba, Spain
Javier Sánchez-Monedero	University of Córdoba, Spain
Manuel Cruz-Ramírez	University of Córdoba, Spain
Ricardo Sousa	INESC, Portugal
Arie Ben David	University of Córdoba, Spain
David Becerra-Alonso	University of Córdoba, Spain

Hybrid Metaheuristics for Combinatorial Optimization and Modelling Complex Systems

José RamónVillar	University of Oviedo, Spain
Camelia Chira	University of Babes-Bolyai, Romania
Enrique de la Cal	University of Oviedo, Spain
Anca Gog	University of Babes-Bolyai, Romania
Camelia Pintea	North University Baia-Mare, Romania
Gerardo Méndez	Instituto Tecnológico Nuevo León, Mexico
Javier Sedano	Instituto Tecnológico de Castilla y León, Spain
José Luis Calvo Rolle	University of Coruña, Spain
Petrica Pop	North University Baia-Mare, Romania
Adolfo Rodríguez	University of León, Spain
María Sierra	University of Oviedo, Spain
Óscar Ibañez	European Centre of Soft Computing, Spain
André Carvalho	University of São Paulo, Brazil
Luciano Sánchez	University of Oviedo, Spain
Paola Mello	University of Bologna, Italy
Nima Hatami	University of Cagliari, Italy

Hybrid Computational Intelligence and Lattice Computing for Image and Signal Processing

Manuel Graña	University of the Basque Country/EHU, Spain
Alexandre Savio	University of the Basque Country/EHU, Spain
Borja Fernandez-Gauna	University of the Basque Country/EHU, Spain

Darya Chyzhyk	University of the Basque Country/EHU, Spain
Ekaitz Zulueta	University of the Basque Country/EHU, Spain
Ion Marques	University of the Basque Country/EHU, Spain
Josu Maiora	University of the Basque Country/EHU, Spain
Miguel Ángel Veganzones	University of the Basque Country/EHU, Spain
Ana I Gonzalez	University of the Basque Country/EHU, Spain
Dragan Simic	University of Novi Sad, Serbia
Iñigo Barandiaran	Vicomtech, Spain
Israel Rebollo Ruiz	University of the Basque Country/EHU, Spain
Maite Termenon	University of the Basque Country/EHU, Spain
Ivan Macia	Vicomtech, Spain
Borja Ayerdi	University of the Basque Country/EHU, Spain
Elsa Fernández	University of the Basque Country/EHU, Spain
Andoni Beristain	Vicomtech, Spain
Ramón Moreno	University of the Basque Country/EHU, Spain

Workshop Committees

Nonstationary Models of Pattern Recognition and Classifier Combinations

Michał Woźniak	Wroclaw University of Technology, Poland
Emilio Corchado	University of Salamanca, Spain
Boguslaw Cyganek	AGH University of Science and Technology, Poland
Francisco Herrera	University of Granada, Spain
Giorgio Fumera	University of Cagliari, Italy
Ioannis Katakis	University of Cyprus, Greece
Manuel Graña	University of the Basque Country/EHU, Spain
Robert Burduk	Wroclaw University of Technology, Poland
Jerzy Stefanowski	Poznan University of Technology, Poland
Przemysław Kazienko	Wroclaw University of Technology, Poland
Álvaro Herrero	University of Burgos, Spain
Bruno Baruque	University of Burgos, Spain
Piotr Sobolewski	Wroclaw University of Technology, Poland
Konrad Jackowski	Wroclaw University of Technology, Poland
Václav Snášel	VSB-Technical University of Ostrava, Poland
Piotr Cal	Wroclaw University of Technology, Poland
Marcin Zmyślony	Wroclaw University of Technology, Poland
Konstantinos Sirlantzis	University of Kent, UK

Organizing Committee

Emilio Corchado	University of Salamanca, Spain (Co-chair)
Bruno Baruque	University of Burgos, Spain (Co-chair)
Álvaro Herrero	University of Burgos, Spain (Co-chair)

José Luis Calvo	University of Coruña, Spain (Co-chair)
Leticia Curiel	University of Burgos, Spain
M^a Dolores Muñoz	University of Salamanca, Spain
Ángel Arroyo	University of Burgos, Spain
Javier Sedano	University of Burgos, Spain
Fernando De la Prieta	University of Salamanca, Spain
Ana Gil	University of Salamanca, Spain
M^a Araceli Sánchez	University of Salamanca, Spain
Héctor Quintián	University of Salamanca, Spain
Héctor Casado	University of Salamanca, Spain
Antonio J. Sánchez	University of Salamanca, Spain

Table of Contents – Part II

Special Sessions

Methods of Classifier Fusion

HAIS for Computer Security (HAISFCS)

Data Mining: Data Preparation and Analysis

Hybrid Artificial Intelligence Systems in Management of Production Systems

Hybrid Artificial Intelligent Systems for Ordinal Regression

Hybrid Metaheuristics for Combinatorial Optimization and Modelling Complex Systems

Hybrid Computational Intelligence and Lattice Computing for Image and Signal Processing

Workshop

Nonstationary Models of Pattern Recognition and Classifier Combinations

Table of Contents – Part I

Special Sessions

Agents and Multi Agents Systems

HAIS Applications

Cluster Analysis

Data Mining and Knowledge Discovery

Evolutionary Computation

Learning Algorithms

Systems, Man, and Cybernetics by HAIS

Hybrid Decision Tree Architecture Utilizing Local SVMs for Multi-Label Classification

Gjorgji Madjarov and Dejan Gjorgjevikj

Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Skopje, Macedonia
{gjorgji.madjarov,dejan.gjorgjevikj}@finki.ukim.mk

Abstract. Multi-label classification (MLC) problems abound in many areas, including text categorization, protein function classification, and semantic annotation of multimedia. Issues that severely limit the applicability of many current machine learning approaches to MLC are the large-scale problem and the high dimensionality of the label space, which have a strong impact on the computational complexity of learning. These problems are especially pronounced for approaches that transform MLC problems into a set of binary classification problems for which SVMs are used. On the other hand, the most efficient approaches to MLC, based on decision trees, have clearly lower predictive performance. We propose a hybrid decision tree architecture that utilizes local SVMs for efficient multi-label classification. We build decision trees for MLC, where the leaves do not give multi-label predictions directly, but rather contain SVM-based classifiers giving multi-label predictions. A binary relevance architecture is employed in each leaf, where a binary SVM classifier is built for each of the labels relevant to that particular leaf. We use several real-world datasets to evaluate the proposed method and its competition. Our hybrid approach on almost every classification problem outperforms the predictive performances of SVM-based approaches while its computational efficiency is significantly improved as a result of the integrated decision tree.

Keywords: multi-label classification, hybrid architecture.

1 Introduction

Single-label classification is concerned with learning from examples, each associated with a single label λ_i from a finite set of disjoint labels $L = \{\lambda_1, \lambda_2, ..., \lambda_Q\}$, $Q > 1$. The task is to learn a mapping l: $X \rightarrow L$ (where X denotes the example space) that assigns a single label to each example. For $Q > 2$, the task is referred to as a *multi-class classification*.

In multi-label classification (MLC), the task is to learn a mapping m: $X \rightarrow 2^L$. Each example $x \in X$ is associated to a set of labels $Y \subseteq L$. In contrast to multi-class classification, the labels are not assumed to be mutually exclusive, i.e., an example can be a member of more than one class. The labels in Y are called relevant and those in $L \backslash Y$ irrelevant for a given example.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 1–12, 2012.

Two major classes of algorithms for multi-label classification are decision-tree-based and SVM-based approaches. The first group is extremely efficient, but not very accurate while the second group, represented by the problem transformation SVM architectures [14] are very accurate, but can be computationally expensive, especially when labels abound. In this paper, we propose a novel hybrid architecture that integrates Decision Trees and Support Vector Machines for computationally efficient multi-label classification (ML-SVMDT) on large-scale problems with a large number of labels.

Section 2 surveys related previous work in multi-label learning. The architecture that we propose is presented in Section 3. Section 4 presents the experimental results that compare the performance of our architecture with the competing methods. The conclusions are given in Section 5.

2 Related Work

2.1 The Landscape of MLC Approaches

The issue of learning from multi-label data has recently attracted significant attention from many researchers. They are motivated by an increasing number of new applications, such as semantic annotation of images and video, functional genomics, music categorization into emotions, text classification, directed marketing and others. Many different approaches have been developed to solve the multi-label learning problems. Tsoumakas et al.[14] summarize them into two main categories: a) algorithm adaptation methods, and b) problem transformation methods. Algorithm adaptation methods extend specific learning algorithms to handle multi-label data directly. Examples include lazy learning [18], decision trees [2], neural networks [17], boosting [11], etc.

ML-C4.5 [2] is an adaptation of the well known C4.5 algorithm for multi-label learning. Clare et al. modified the formula of entropy calculation (equation 1) in order to solve multi-label problems. The modified entropy sums the entropies for each individual class label.

$$entropy(S) = -\sum_{i=1}^{N} (p(c_i)\log p(c_i) + q(c_i)\log q(c_i)) \tag{1}$$

where S is the set of examples, $p(c_i)$ is the relative frequency of class label c_i and $q(c_i) = 1 - p(c_i)$. They also allowed multiple labels in the leaves of the tree. Each leaf is represented by the most frequent set of class labels that are associated to the training examples that belong to that leaf. If more than 50% of the training examples in the leaf are labeled with a particular label then that label belongs to the set of most frequent labels.

ML-kNN [18] is based on the popular k Nearest Neighbors (kNN) lazy learning algorithm. The first step in this approach is the same as in kNN, i.e., retrieving the k nearest examples. It uses the maximum a posteriori principle in order to determine the label set of the test example, based on prior and posterior probabilities i.e. the frequency of each label within the k nearest neighbors.

Problem transformation methods, on the other hand, transform the multi-label learning problem into one or more single-label classification problems. The simplest and the most efficient strategy in terms of computational complexity in the multi-label setting is the one-against-all strategy, also referred to as the binary relevance (BR) method [14]. It addresses the multi-label learning problem by learning Q binary classifiers - one classifier for each label L. It transforms the original data set into Q data sets $S_{\lambda_i}, i = 1...Q$ that contain all examples of the original data set, labeled positively if the label set of the original example contained λ_i and negatively otherwise. For the classification of a new instance, each binary classifier predicts whether its label λ_i is relevant for the given example or not. Actually, BR outputs the union of the labels λ_i that are positively predicted by the Q classifiers. In the ranking scenario, the labels are ordered according to the probability associated to each label by the respective binary classifier.

A method closely related to the BR method is the Classifier Chain method (CC) proposed by Read et al. [10]. This method involves Q binary classifiers as in BR. Classifiers are linked along a chain where each classifier deals with the binary relevance problem associated with label $\lambda_i \in L$. The feature space of each link in the chain is extended with the 0/1 label associations of all previous links.

HOMER (Hierarchy Of Multi-label classifiERs) [15] is a computationally efficient multi-label classification method specifically designed for large multi-label datasets. HOMER constructs a hierarchy of multi-label classifiers, each one dealing with a much smaller set of labels compared to Q (the total number of labels) and a more balanced example distribution. This leads to improved predictive performance and also to linear training and logarithmic testing complexities with respect to Q. One of the main processes within HOMER is the even distribution of a set of labels into k disjoint subsets so that similar labels are placed together and dissimilar apart. The best predictive performance is reported using a balanced k means algorithm customized for HOMER [15].

Recently the most challenging issues in MLC are the high dimensionality of the label space and the problem of large datasets. These two problems can significantly influence on the computational complexity and the predictive performance of the MLC methods. Some proposed methods achieve higher computational efficiency at the cost of predictive accuracy, such as ML-kNN [18], ML-C4.5 [2], etc. These methods usually belong to the group of algorithm adaptation methods. Other proposed methods, based on problem transformation, use base classifiers with higher computational efficiency, such as Naive Bayes [7] [15], the one-layer perceptron [9], etc., in order to reduce the computational complexity.

2.2 Combining Decision Trees and SVMs

Several approaches that combine decision trees and SVMs have been proposed for binary and multi-class classification. For example, Kumar et al. [8] propose a method that combines decision trees and global SVM models (models learned on the whole dataset) for solving binary classification problems. Other approaches, such as [3], use the structure of the decision tree to organize/arrange SVM classifiers in its nodes in order to improve the computational efficiency and the

Table 1. The process of building the ML-SVMDT

procedure ML-SVMDT(S) **returns** tree
1: $(f^*, g^*, \mathcal{P}^*) = \text{BestFeature}(S)$
2: **if** $f^* \neq none$ **then**
3: **for each** $S_k \in \mathcal{P}^*$ **do**
4: $tree_k = \text{ML-SVMDT}(S_k)$
5: **return** node(f^*, $\bigcup_k \{tree_k\}$)
6: **else**
7: $localSVM = \text{trainBRModel}(S)$
8: **return** leaf($localSVM$)

procedure BestFeature(S)
1: $(f^*, g^*, \mathcal{P}^*) = (none, 0, \emptyset)$
2: **for each** feature f **do**
3: $\mathcal{P} =$ partition induced by f on S
4: $g = entropy(S) - \sum_{S_k \in \mathcal{P}} \frac{|S_k|}{|S|} entropy(S_k)$
5: **if** $(g > g^*) \wedge \text{Acceptable}(f, \mathcal{P})$ **then**
6: $(f^*, g^*, \mathcal{P}^*) = (f, g, \mathcal{P})$
7: **return** $(f^*, g^*, \mathcal{P}^*)$

predictive performance. Boosting ensemble of support vector machines for multi-class classification was proposed in [13]. Gama [5] proposed functional trees for multi-class classification and regression problems. However, none of these approaches deal with MLC problems.

3 Integration of Decision Trees and SVMs

In this paper, we propose a novel hybrid approach for computationally efficient multi-label classification that combines the algorithm adaptation method ML-C4.5 [2] and the problem transformation method Binary Relevance (BR) [14]: The latter uses SVMs as base classifiers for solving the partial binary classification problems. The main idea of our approach is to use the advantages of both methods - the low computational complexity of ML-C4.5 and the predictive accuracy of the BR architecture with SVM classifiers.

One approach to achieve effective and computationally efficient multi-label classification is to partition the global classification problem first and then learn local classifiers for each of those partitions (subproblems) separately. In the prediction phase, first one tries to determine the partition to which a test example belongs, and then to classify the example using a local classifier trained using the examples belonging to that partition only. The logic behind this approach is that the "neighbors" of a test example (training examples that belong to the same partition as the test example), would be able to provide more accurate information about it faster.

We propose a novel hybrid architecture that introduces local models for solving multi-label learning problems, based on SVM classifiers. Our approach combines the ML-C4.5 method for partitioning the input feature space and the BR method utilizing SVMs as base classifiers for local classification. The main idea is to use the advantages of both methods, in order to build an architecture that will improve the predictive performance and the computational efficiency.

Throughout the literature, BR is often mentioned, but consistently criticized on account of its assumption of label independence. Namely, during the transformation process, BR ignores label correlations that exist in the training data. In order to reduce these inconsistencies in the multi-label prediction of the BR

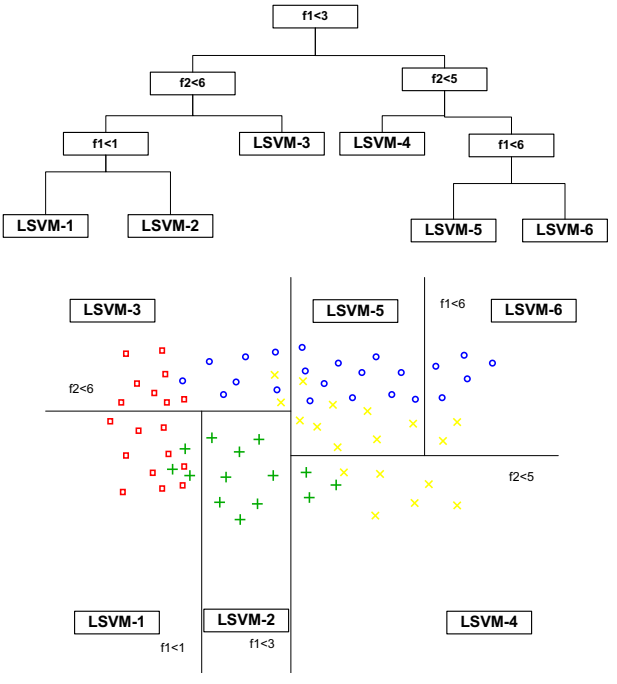


Fig. 1. ML-SVMDT splits the original dataset into subsets and builds a local SVM model (LSVM) for each partition

method, we employed the ML-C4.5 method. In this context, the ML-C4.5 method actually tries to separate the training examples in groups where the label inconsistency will be eliminated. It finds the correlation between labels and according to that correlation it splits the global problem into several local subproblems.

The procedure (ML-SVMDT(S)) of building the architecture is outlined in Table 1. It takes as input a set of examples (S) and outputs a tree. The process of building starts at the root of the tree with choosing one feature (f) of the data that most effectively splits the dataset of examples into subsets ($\mathcal{P}$). The criterion is the normalized information gain (difference in entropy g) that results from choosing a feature for splitting the data (line 6 of BestFeature procedure in Table 1). The feature with the highest normalized information gain is chosen to make the decision. The process continues recursively in each node until all the examples are labeled with the same labels or none of the features provide any information gain. Tree building also stops if some predetermined minimal number of examples per node is reached. If no acceptable feature can be found or some predetermined minimal number of examples per node is reached (Acceptable($f, \mathcal{P}$)), the process of splitting the dataset stops and the corresponding node is declared as a leaf. After that, we try to solve the "new" problem defined on the examples in the leaf by using a local model. This means that every leaf of the decision

tree is replaced with one local model built by the BR method on the training examples that belong to the corresponding leaf, using SVMs as base classifiers (Figure 1).

The testing for each test example starts at the root of the tree. The decision tree transfers the example to exactly one leaf of the tree according to its features. The final decision about the labeling of the test example is performed by the local model in the corresponding node consisting of SVMs. Each test example consults only one local model in order to be classified. The testing time for each test example is the sum of the time needed to sort the example through the decision tree and the time needed for the corresponding local model to make a decision.

4 Experiments

In this section the performance of the proposed method is compared to the performances of the competing method for multi-label learning in domains with large number of labels (HOMER [15]) and two additional state of the art methods (Classifier Chains - CC [10] and ML-kNN [18]). We also compare it to the two methods that it integrates, the ML-C4.5 [2] method and the BR [14] method. The performance is measured in terms of five different multi-label evaluation measures (two example based measures - Hamming loss and precision, two label based measures - micro precision and macro precision and one ranking based measure - average precision) [16]. In the results we also report the training and the testing times of all methods (measured in seconds).

4.1 Datasets and Experimental Setup

Five different multi-label classification problems were addressed in our experiments. The predictive performance in terms of the measures mentioned above and the training and testing times were recorded for every method for each classification problem. The complete description of the datasets in terms of application domain (*domain*), the number of training (*#tr.e.*) and test (*#t.e.*) examples, the number of features (D), the total number of labels (Q) and label cardinality (l_c) [14] are shown in Table 2.

We strived to include a considerable variety and scale of multi-label datasets. In total we use five datasets, with dimensions ranging from 101 to 983 labels,

Table 2. Dataset description

	domain	*#tr.e.*	*#t.e.*	D	Q	l_c
corel5k [4]	image	4500	500	499	374	3.52
mediamill [12]	video	30993	12914	120	101	4.38
bibtex [7]	text	4880	2515	1836	159	2.40
delicious [15]	text	12920	3185	500	983	19.02
bookmarks [7]	text	60000	27856	2150	208	2.03

and from less than 5,000 examples to almost 90,000. The datasets are roughly ordered by complexity ($\#tr.e. \times D \times Q$).

The training and testing of the proposed method were performed using a custom developed application that uses the MULAN library[1] for the machine learning framework Weka [6]. We implemented the ML-C4.5 method within the same library, while HOMER and BR was already implemented in MULAN. For the CC method we used the MEKA[2] extension for the WEKA framework. All experiments were performed on a server with an Intel Xeon processor at 2.50GHz on 64GB of RAM with the Fedora 14 operating system.

The LIBSVM library [1], and in particular SVMs with a radial basis kernel, were used for solving the binary classification problems for all datasets in the BR, CC, HOMER and the proposed method. The kernel parameter *gamma* and the penalty C for the datasets for each method were determined by 10-fold cross validation using only the training sets. The values $2^{-15}, 2^{-13}, ..., 2^{1}, 2^{3}$ were consider for *gamma* and $2^{-5}, 2^{-3}, ..., 2^{13}, 2^{15}$ for the penalty C. After determining the best parameters values for each method on every dataset the classifiers were trained using all available training examples and were evaluated by recognizing all test examples from the corresponding dataset.

The ML-C4.5 method uses subtree raising with a pruning confidence of 0.25 as a post pruning strategy in all classification problems. The number of neighbors in the ML-kNN method for each dataset was determined from the values 6 to 20 with step 2 for which the best results were obtained. To define the subsets of labels in each level of the hierarchy in HOMER, we used the balanced k means algorithm proposed by the original authors. This algorithm requires one parameter to be configured: number of subsets k. Five different values (2-6) were considered in the experiments for this parameter. These values were used by the authors [15]. For all methods, the best obtained results are presented in the Results subsection. Additionally, to access the dependence of the predictive performance and the computational complexity of ML-SVMDT on the minimal number of examples in the leaves of the decision tree, we tried six different values of the minimal number of examples (30-80) in the leaves. Overall, the proposed architecture obtained the best predictive performance when the minimal number of examples in the leaves was set to 70 and these results are presented in the Results subsection.

4.2 Results

Table 3 gives the predictive performance in terms of the five evaluation measures, the training and testing times for each method on each of the datasets. The first column of the table lists the evaluation measures, while the second column lists the classification problems. The remaining columns show the performance of each method for every dataset. The best results per dataset in the table are shown in boldface. DNF in the results indicates that the experiment Did Not Finish

[1] http://mulan.sourceforge.net/
[2] http://meka.sourceforge.net/

Table 3. The predictive performance in terms of the five evaluation measures, the training and the testing times measured in seconds

	datasets	BR	CC	ML-C4.5	ML-kNN	HOMER	ML-SVMDT
Hamming loss	**corel5k**	0.017	0.017	0.010	**0.009**	0.012	**0.009**
	mediamill	0.032	0.032	0.044	**0.031**	0.038	0.032
	bibtex	**0.012**	**0.012**	0.016	0.014	0.014	**0.012**
	delicious	**0.018**	**0.018**	0.019	**0.018**	0.022	**0.018**
	bookmarks	DNF	DNF	**0.009**	**0.009**	DNF	**0.009**
Precision	**corel5k**	0.042	0.042	0.005	0.035	**0.317**	0.127
	mediamill	0.731	**0.741**	0.056	0.724	0.597	0.727
	bibtex	**0.515**	0.508	0.123	0.254	0.472	0.484
	delicious	0.443	0.399	0.001	0.424	0.369	**0.486**
	bookmarks	DNF	DNF	0.271	0.218	DNF	**0.281**
mi. Precision	**corel5k**	0.061	0.061	0.160	**0.730**	0.308	0.664
	mediamill	0.742	**0.753**	0.597	0.739	0.569	0.749
	bibtex	0.753	0.744	0.359	**0.819**	0.547	0.789
	delicious	0.658	0.660	0.000	0.651	0.396	**0.662**
	bookmarks	DNF	DNF	0.632	0.850	DNF	**0.855**
ma. Precision	**corel5k**	0.052	0.053	0.004	0.031	0.044	**0.055**
	mediamill	0.112	0.144	0.046	**0.308**	0.107	0.258
	bibtex	0.528	**0.539**	0.128	0.192	0.391	0.495
	delicious	0.299	0.303	0.000	0.134	0.154	**0.312**
	bookmarks	DNF	DNF	0.292	0.414	DNF	**0.485**
Avg. Precision	**corel5k**	0.303	0.293	0.196	0.266	0.222	**0.306**
	mediamill	0.686	0.672	0.669	**0.703**	0.583	0.698
	bibtex	0.597	**0.599**	0.392	0.349	0.407	0.563
	delicious	0.351	0.343	0.321	0.326	0.231	**0.362**
	bookmarks	DNF	DNF	0.378	0.381	DNF	**0.421**
Training times	**corel5k**	926	1225	369	389	771	**274**
	mediamill	85468	100435	2030	**1094**	78195	9015
	bibtex	11013	12434	566	**124**	2869	767
	delicious	57053	84903	2738	**236**	21218	1168
	bookmarks	DNF	DNF	**4039**	15990	DNF	53737
Testing times	**corel5k**	25	31	**1**	45	14	9
	mediamill	6152	6125	**1**	477	6079	398
	bibtex	654	661	**7**	64	155	84
	delicious	2045	1872	**19**	55	816	189
	bookmarks	DNF	DNF	**21**	4084	DNF	4189

within one week under the available resources. Training time of the ML-kNN method is the time needed for calculating the posterior probability of each label within the k nearest neighbors.

Overall, the results show that ML-SVMDT outperforms HOMER and ML-C4.5 on all five datasets in terms of the five evaluation measures. The difference in the predictive performances between ML-SVMDT and HOMER is more evident for the larger datasets (bibtex and delicious). Compared to the BR, CC and the ML-kNN methods, ML-SVMDT shows better performance in terms of the ranking based measure for all datasets, except for the bibtex dataset where BR and CC show slightly better results. For the example and label based measures the proposed method outperforms BR, CC and ML-kNN for the delicious dataset and shows comparable performance for the corel5k, mediamill and bibtex datasets.

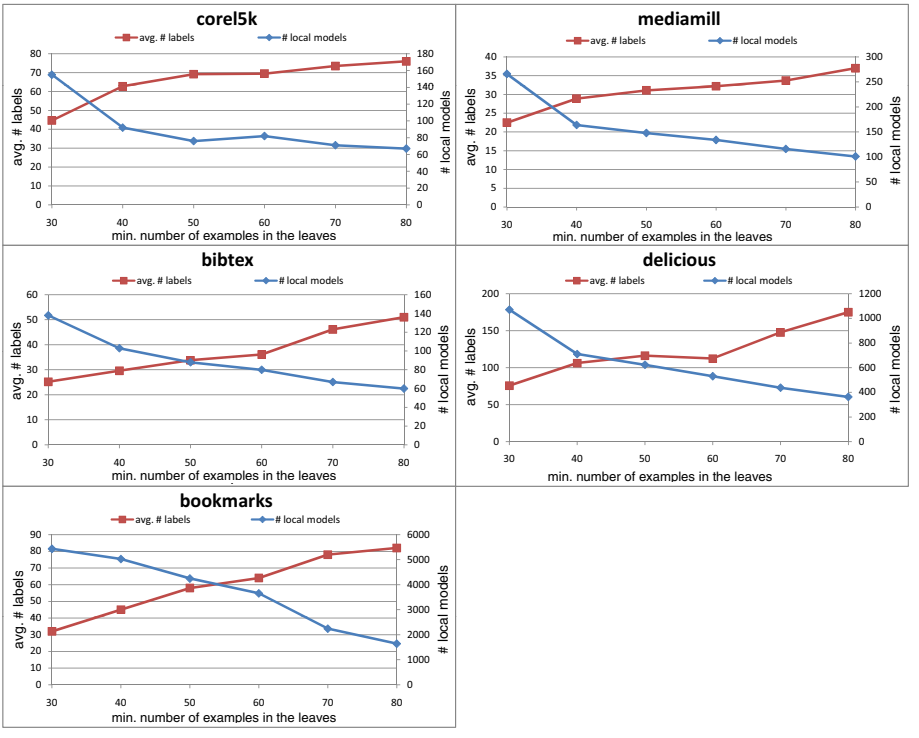
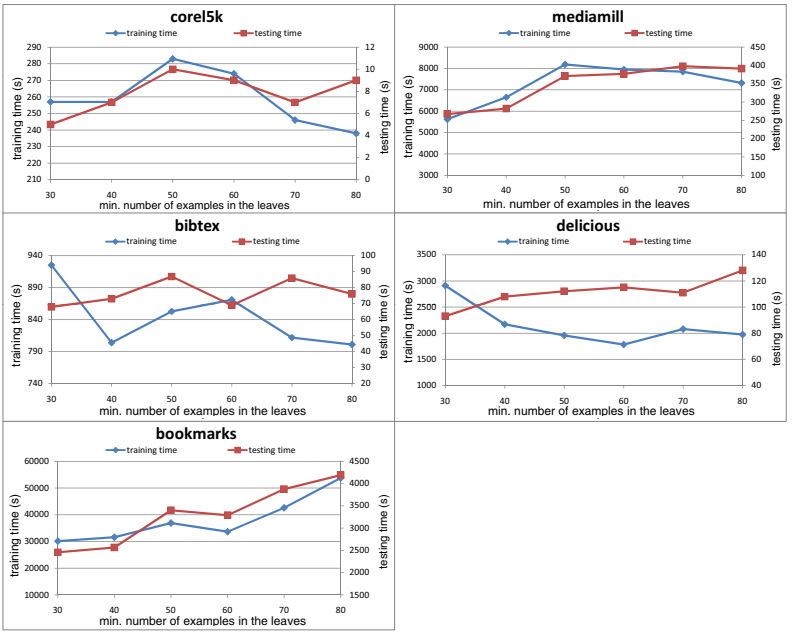
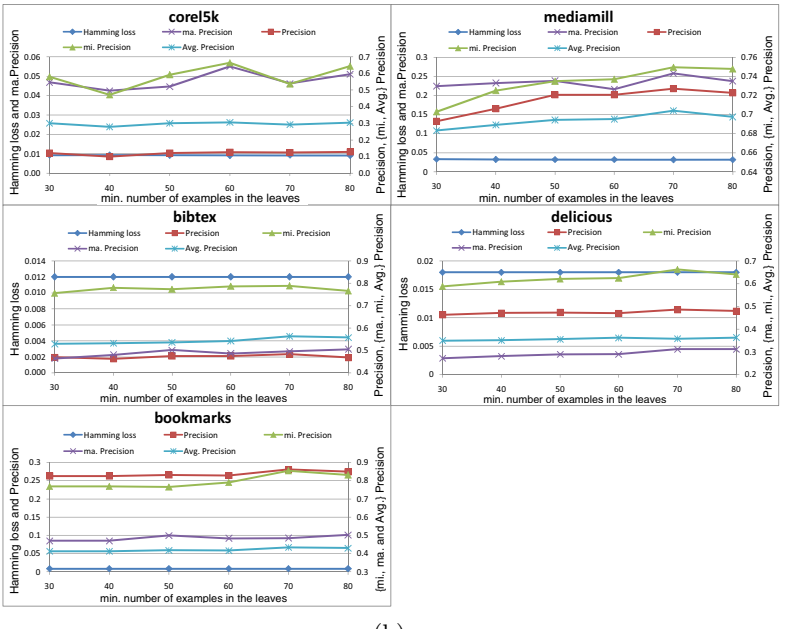


Fig. 2. The average number of labels per local model and the number of local models of ML-SVMDT as functions of the number of the minimal number of examples in the leaves

The results in terms of training and testing speed show that the proposed method is 2 to 20 times faster in the training phase and 1.4 to 10 times faster in the testing phase than the HOMER method. The computational efficiency of ML-SVMDT is even higher compared to the BR and CC methods (5 to 35 times in the training phase and 3 to 25 times in the testing phase). ML-kNN is the fastest in the training phase for the mediamill, bibtex and delicious datasets, while ML-C4.5 shows the best training time for the bookmarks dataset. The proposed architecture also shows higher computational efficiency than the ML-C4.5 method in the training phase for the datasets with a larger number of labels (corel5k and delicious) as a result of the post-pruning method used in the ML-C4.5 algorithm that gives additional computational complexity. ML-SVMDT uses only the minimal number of examples in the leaves of the tree as a pre-pruning method that controls the size of the tree: After a node reaches the minimal number of examples, no further branching of the decision tree is allowed. On the other hand, ML-C4.5 is faster than ML-SVMDT in the training phase for the other three datasets (mediamill, bibtex and bookmarks) and it is the fastest in the testing phase for all datasets.



(a)



(b)

Fig. 3. (a) The training and testing times and (b) the predictive performance of ML-SVMDT as functions of the number of the minimal number of examples in the leaves

In the case of the bookmarks dataset, only the ML-C4.5, ML-kNN and the ML-SVMDT methods perform satisfactorily under the experiment setup. The other methods suffer problem of higher computational complexity. Bookmarks dataset is much larger than those typically approached in the literature. On this dataset, ML-SVMDT shows the best predictive performance and the overall advantage in time costs compared to the other methods.

The dependence of the predictive performance and the training and testing times of the proposed architecture on different values of the minimal number of examples in the leaves graphically are shown on Figures 3(a) and 3(b), for each dataset separately. It should be noticed that the predictive performance of the proposed architecture strongly depends on this parameter. Overall, the best predictive performance were obtained when the minimal number of examples in leaves was set to 70. For the bookmarks dataset, some predictive performance further improved when the minimal number of examples in leaves rose to 80 as a result of the larger number of examples in the dataset compared to the other datasets. On Figure 2, the dependence of the average number of labels per local model (avg. # labels) and the number of local models (# local models) generated in the ML-SVMDT architecture on the minimal number of examples in the leaves are shown graphically for each dataset. These two parameters are closely related to the number of examples in the leaves: By increasing the minimal number of examples in the leaves, the average number of labels per local model increases, while the number of local models generated by ML-SVMDT decreases.

5 Conclusions

We propose a novel hybrid architecture that integrates Decision Trees and SVMs for computationally efficient multi-label classification. The architecture combines the algorithm adaptation method ML-C4.5 and the problem transformation method Binary Relevance: The latter uses SVMs as base classifiers for solving the binary classification problems.

The proposed architecture is compared to the BR method, CC, ML-C4.5, ML-kNN and the HOMER method on five different real-world datasets. Among the six competing methods, ML-C4.5 is the fastest one in the prediction phase. ML-kNN shows lower computational complexity in the training and slightly higher computational complexity in the prediction phase compared to the ML-C4.5 method, but, it is better in terms of the predictive performance. ML-SVMDT shows slightly higher training times and comparable, but slightly smaller testing times than ML-kNN, while showing better predictive performance in terms of the example and ranking based evaluation measures. In terms of the label based measures ML-SVMDT outperforms the ML-kNN method for the two largest datasets (delicious and bookmarks) and corel5k (the second dataset ordered by label dimensionality) and shows comparable results for the other two datasets. Compared to the BR and CC methods, it shows slightly better predictive performance but significantly higher computational efficiency. ML-SVMDT also clearly outperforms the HOMER method in both predictive performance and speed.

Despite other methods, ML-SVMDT can achieve better predictive performance and is efficient enough to scale up to very large problems.

References

1. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
2. Clare, A., King, R.D.: Knowledge Discovery in Multi-label Phenotype Data. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 42–53. Springer, Heidelberg (2001)
3. Dong, G.M., Chen, J.: Study on support vector machine based decision tree and application. In: Proc. of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 318–322 (2008)
4. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.: Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
5. Gama, J.: Functional trees. Machine Learning 55, 219–250 (2004)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explorations 11, 10–18 (2009)
7. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel Text Classification for Automated Tag Suggestion. In: Proc. of the ECML/PKDD Discovery Challenge (2008)
8. Kumar, A.M., Gopal, M.: A hybrid svm based decision tree. Pattern Recognition 43, 3977–3987 (2010)
9. Mencía, E.L., Park, S.H., Fürnkranz, J.: Efficient voting prediction for pairwise multilabel classification. Neurocomputing 73, 1164–1176 (2010)
10. Read, J., Pfahringer, B., Holmes, G.: Multi-label Classification Using Ensembles of Pruned Sets. In: Proc. of the 8th IEEE International Conference on Data Mining, pp. 995–1000 (2008)
11. Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning 39, 135–168 (2000)
12. Snoek, C.G.M., Worring, M., van Gemert, J.C., Geusebroek, J.M., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: Proc. of the 14th Annual ACM International Conference on Multimedia, pp. 421–430 (2006)
13. Ting, K.M., Zhu, L.: Boosting Support Vector Machines Successfully. In: Benediktsson, J.A., Kittler, J., Roli, F. (eds.) MCS 2009. LNCS, vol. 5519, pp. 509–518. Springer, Heidelberg (2009)
14. Tsoumakas, G., Katakis, I.: Multi Label Classification: An Overview. International Journal of Data Warehouse and Mining 3(3), 1–13 (2007)
15. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In: Proc. of the ECML/PKDD Workshop on Mining Multidimensional Data, pp. 30–44 (2008)
16. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer, Heidelberg (2010)
17. 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(10), 1338–1351 (2006)
18. Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition 40(7), 2038–2048 (2007)

Ensemble Pruning Using Harmony Search

Shina Sheen[1,*], S.V. Aishwarya[1], R. Anitha[1], S.V. Raghavan[2], and S.M. Bhaskar[3]

[1] PSG College of Technology, Coimbatore, India
shina_np12@yahoo.com, aishwarya.sv@gmail.com,
anitha_nadarajan@mail.psgtech.ac.in
[2] Indian Institute of Technology, Madras
svr@cs.iitm.ernet.in
[3] CCA&R, New Delhi
smb@nic.in

Abstract. In recent years, a number of works proposing the combination of multiple classifiers to produce a single classification have been reported. The resulting classifier, referred to as an ensemble classifier, is generally found to be more accurate than any of the individual classifiers making up the ensemble. In an ensemble of classifiers, it is hoped that each individual classifier will focus on different aspects of the data and error under different circumstances. By combining a set of so-called base classifiers, the deficiencies of each classifier may be compensated by the efficiency of the others. Ensemble pruning deals with the reduction of an ensemble of predictive models in order to improve its efficiency and performance. Ensemble pruning can be considered as an optimization problem. In our work we propose the use of Harmony search, a music inspired algorithm to prune and select the best combination of classifiers. The work is compared with AdaBoost and Bagging among other popular ensemble methods and our method is shown to perform better than the other methods. We have also compared our work with an ensemble pruning technique based on genetic algorithm and our model has shown better accuracy.

Keywords: Ensemble learning, Ensemble pruning, Harmony search, Classification.

1 Introduction

There is a large body of theoretical and empirical research showing that combining the predictions of a collection of classifiers can be an effective strategy to improve generalization performance [1-4]. The combination methods proposed in the literature are based on "voting" rules, statistical techniques, belief functions, and other "classifier fusion" schemes. As an example, the "majority" voting rule interprets each classification result as a "vote" for one of the data classes and assigns the input pattern to the class receiving the majority of votes. Such methods assume that, for each pattern, different classifiers make different classification errors. Ensembles

* Corresponding author.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 13–24, 2012.

composed of independent classifiers generally use equally weighted voting to produce a final class prediction. Given an unlabeled instance, the usual procedure is to query all classifiers in the ensemble and then output the majority class. In this work, we show that it is possible to estimate the outcome of the voting process with a specified confidence level by polling only a subset of the classifiers in the ensemble. We make use of Harmony search, a music inspired algorithm to identify the best subset of classifiers for a specific task. Using this procedure, only a subset of the predictors in the ensemble needs to be queried, which results in significant speed up of the classification process.

The ensemble method we propose in this paper comprises of three phases: the generation of multiple predictive models or classifiers, reduction of ensemble size prior to combination called as ensemble pruning and the combination of the final ensemble.

2 Ensemble of Classifiers

Classification is one task of data mining which allows predicting if a data instance is a member of a predefined class. Input is a training dataset S, where each instance is typically represented in the form of vector attributes $<x_1,x_2,x_3\ldots,x_n,y>$, y is the class attribute. The objective of classification is to train a classification algorithm A, on training data set S to find a good approximation of a certain function f(x)=y which is called the classifier. Evaluation of classifier accuracy is performed with a dataset T independent of S. The classifier will thereafter be able to predict the class y for new data d. An ensemble contains a number of classifiers called base learners. The generalization ability of an ensemble is usually much stronger than that of base learners. The various phases in an ensemble method is presented below.

2.1 Generating Models

Typically, an ensemble is constructed in two steps. First, a number of base learners are produced, which can be generated in a *parallel* style or in a *sequential* style where the generation of a base learner has influence on the generation of subsequent learners. Then, the base learners are combined to be used for prediction, where the most popular combination schemes are *majority voting* for classification and *weighted averaging* for regression. The basic principle in ensemble learning is to generate multiple versions of the classifier by perturbing the training set, construction method or some parameters. Several techniques have been used for this and the most notable among these are bagging[2], boosting[5] and random subspace method[7].

An ensemble can be composed of homogeneous or heterogeneous models. Our ensemble model comprises of heterogeneous models derived from running different learning algorithms on the same data set.

2.2 Pruning the Ensemble

Let $\Omega= \{M_1, \ldots..M_n\}$ be an ensemble of n classifiers. M_i is a classifier that can predict the class $M_i(x)$ of an observation x. The problem of ensemble pruning is to find the

best subset of Ω such that the combination of the selected classifiers will have the highest possible degree of accuracy.

The various ensemble pruning methods in literature generally fall into the following categories: Ranking based [8-11] and search based [12-15]. Prodromidis et al [8] suggested ranking classifiers according to their classification performance on a separate validation set and their ability to correctly classify specific classes. Similarly Caruana et al [9] presented a forward stepwise selection procedure in order to select the most relevant classifiers (that maximize the ensemble's performance) among thousands of classifiers. The algorithm FS-PP-EROS generates a selective ensemble of rough subspaces [10]. The algorithm performs an accuracy-guided forward search to select the most relevant members. Margineantu and Dietterich [11] presented an agreement based ensemble pruning which measures the Kappa statistics between any pair of classifiers. Pairs of classifiers are then selected in ascending order of their agreement level till the desired ensemble size is reached.

Rokach et al[12] suggested ranking the classifiers first according to their ROC performance and then evaluating the performance of the ensemble subset by using the top ranked members. Prodromidis and Stolfo [13] introduced a backwards correlation based pruning. The main idea is to remove the members that are least correlated to a meta-classifier which is trained based on the classifiers' outputs. In each iteration they remove one member and recompute the new reduced meta-classifier (with the remaining members). The meta-classifier in this case is used to evaluate the collective merit of the ensemble. Zhang et al. [14] formulated the ensemble pruning problem as a quadratic integer programming problem to look for a subset of classifiers that has the optimal accuracy-diversity trade-off. Using a semi-definite programming (SDP) technique, they efficiently approximated the optimal solution. The Gasen-b method [15] performed stochastic search in the space of model subsets using a genetic algorithm. The ensemble is represented as a bit string, using one bit for each model. Models are included or excluded from the ensemble depending on the value of the corresponding bit.

A detailed taxonomy of the various approaches is available in [16].

2.3 Combining the Models

Once the classifiers are built, various techniques can be used to combine the results of each classifier. The most cited in literature are the majority vote, the weighted vote and stacking[6]. In majority voting, each classifier outputs a class value and the class with most votes is the one proposed by the ensemble. In weighted voting, the models are not treated equally as each of them is associated with a coefficient (weight), usually proportional to its classification accuracy. Stacking is a method that combines models by learning a meta-level (or level-1) model that predicts the correct class based on the decisions of the base level (or level-0) models. This model is induced on a set of meta-level training data that are typically produced by applying a procedure similar to k-fold cross validation on the training data. The outputs of the base-learners for each instance along with the true class of that instance form a meta-instance. A meta-classifier is then trained on the meta-instances. When a new instance appears for classification, the output of the all base-learners is first calculated and then propagated to the meta-classifier, which outputs the final result.

3 Harmony Search

Harmony search is a music-based metaheuristic optimization algorithm[17-19]. It was inspired by the observation that the aim of music is to search for a perfect state of harmony. This harmony in music is analogous to finding the optimality in an optimization process. A musician always intends to produce a piece of music with perfect harmony. On the other hand, an optimal solution to an optimization problem should be the best solution available to the problem under the given objectives and limited by constraints. Both processes intend to produce the best or optimum.

The key concepts of Harmony Search(HS) algorithm are musicians, notes, harmonies and harmony memory. In most optimisation problems solvable by HS, the musicians are the decision variables of the function being optimised. The notes played by the musicians are the values each decision variable can take. The harmony contains the notes played by all musicians, or a solution vector containing the values for each decision attribute. The harmony memory contains harmonies played by the musicians, or a storage place for solution vectors. A more concrete representation of harmony memory is a two dimensional matrix, where the rows contain harmonies (solution vectors) and the number of rows are predefined and bounded by the harmony memory size. Each column is dedicated to one musician, and the entire column stores all the notes played by him in all harmonies, referred to as the note domain for each musician in this paper. Harmony Search (HS) mimicks the improvisation process of jazz musicians and tries to find the best harmony, i.e., the solution for a certain problem. Consider the problem of optimizing a function $f(x)$ subject to $x_i \in X_i; i = 1,2,\ldots..,n$ where X_i is the possible range for each variable with $x_i^L \leq x_i \leq x_i^U$ where x_i^L and x_i^U are the lower and upper bounds for each variable.

HS works as follows:

Step 1) **Defining the optimization problem and algorithm parameters:** In the first step, the optimization problem is specified as follows:

Minimize (or Maximize) $f(x)$
subjected to $x_i \in X_i$, $i = 1, 2, \ldots, n$.

Step 2) **HM initialization:** In this step, each component of each vector in the parental population (HM), which is of size HMS (Harmony memory size), is filled with random harmonies (solutions) generated according to the bounds of the decision variables x_i.

Step 3) **New harmony improvisation:** In this step, a new harmony vector $\vec{x} = (x_1, x_2, \ldots..., x_n)$ is generated based on three rules: i) memory consideration ii) pitch adjustment and iii) random selection. Generating a new harmony is called 'improvisation.' In the memory consideration, the value of the first decision variable x_1 for the new vector is chosen from any of the values already existing in the current

HM, i.e., from the set $(x_1^1, x_1^2, \ldots\ldots, x_1^{HMS})$, with a probability HMCR. The values of the other decision variables $x_2, \ldots\ldots, x_n$ are also chosen in the same manner. The HMCR, which varies between 0 and 1, is the rate of choosing one value from the previous values stored in the HM, while $(1 - HMCR)$ is the rate of randomly selecting a fresh value from the possible range of values. Every component obtained by the memory consideration is further examined to determine whether it should be pitch adjusted. This operation uses the parameter PAR (which is the rate of pitch adjustment) as follows:

$$x_i = \begin{cases} x_i \pm rand(0,1) \cdot \ bw \ \ with\ probability\ PAR \\ x_i\ with\ probability\ (1 - PAR) \end{cases}$$

where *bw* is an arbitrary distance bandwidth (a scalar number), and *rand*() is a uniformly distributed random number between 0 and 1. Evidently, Step 3 is responsible for generating new potential variation in the algorithm.

Step 4) **HM update:** If the new harmony vector $\vec{x} = (x_1, x_2, \ldots\ldots, x_n)$ is better than the worst harmony in the HM, judged in terms of the objective function value, the new harmony is included in the HM, and the existing worst harmony is excluded from the HM. This is actually the selection step of the algorithm where the objective function value is evaluated to determine if the new variation should be included in the population (HM).

Step 5) **Check stopping criterion:** If the stopping criterion (maximum NI iterations) is satisfied, the computation is terminated. Otherwise, steps 3) and 4) are repeated.

4 Harmony Search for Ensemble Pruning

The aim of this work is to develop a harmony search [18] based, stand alone, reusable search strategy that can find optimal combination of classifiers. We need to map each key concepts of HS into elements in ensemble pruning. There are obvious analogies such as: each classifier combination can be seen as a harmony, and the objective function can be the maximization of accuracy. In this approach we map musicians onto the available classifiers to be selected.

Table 1. Harmony

C1	C2	C3	C4	C5	C6
0	1	1	0	0	0

The note domain of each musician is then a binary value, indicating whether or not the corresponding classifier is included in the harmony and the actual harmony can be represented as a series of bits. For example, as shown in Table 1, a harmony {0,1,1,0,0,0} will translate into classifier subset {C2, C3}.

4.1 Defining the Optimization Problem

Given an ensemble $\Omega = \{M_1, \ldots..M_n\}$, a combination method C, and a training set S from a distribution D over the labeled instance space, the goal is to find an optimal subset $Z_{opt} \subseteq \Omega$ which minimizes the generalization error, over the distribution D of the classifiers in Z_{opt} constructed using method C.

Let $F_1, F_2, \ldots F_n$ be the fitness values measured by the general performance (accuracy) of the classifiers. The problem can be defined as

Maximize $\sum_{i=1}^{n} F_i x_i$
Subject to the constraint $x_i \in (0,1), i = 1,2, \ldots n$

4.2 Initialisation Step

The initialisation step involves filling the harmony memory with randomly generated harmonies, i.e. randomly generated bit sets as shown in table 1. The various parameters are HMS (harmony memory size), PAR(Pitch Adjusting rate) and NI (maximum number of iterations).The harmony memory size is a sensitive parameter. A large harmony memory will give each musician more notes to choose from when improvising a new harmony. However, it will require a longer initialization in order to fill up the harmony memory and hence, may lead to slower convergence.

Table 2. Parameter settings for the proposed model

Parameter	*Value*
HMS	6
PAR	0.1
HMCR	0.9
NI	20

The most significant use of HMCR is in terms of selecting a previously unselected classifier, or vice versa. Pitch adjustment is similar to the mutation operator in genetic algorithms. PAR is usually use 0.1 to 0.5 in most applications. The parameter settings used in our work is shown in Table 2.

4.3 New Harmony Improvisation and Memory Update

A new solution vector is created using the parameters HMCR and PAR. Based on HMCR one of the values in harmony memory is selected or an entirely new value from (0,1) is chosen. The chosen bit is flipped or not based on the value of PAR. If the i^{th} bit of this vector equals 1, then the i^{th} classifier is allowed to participate in classification; if the bit is a 0, then the corresponding classifier does not participate. Each resulting subset of classifiers is evaluated according to its classification accuracy on a set of testing data

using weighted majority voting. In order to improvise a new harmony, each musician randomly selects a value out of their note domain. Together, such selected values form the new bit set. This set is then translated back to a classifier subset and evaluated. If the evaluation score is higher than any of the classifier subsets in the harmony memory, it replaces the worst subset; otherwise, the new bit set is discarded. The process iterates until max iteration is reached.

4.4 Proposed Model

The proposed model is shown in the schematic diagram in Fig 1.The dataset D= $\{(x_i, y_i), i=1,2,,\ldots m)\}$, where x_i is a vector with feature values and y_i is the value of the target variable provided to the n classifiers C_1, C_2,….C_n. Harmony search is used to find out the most optimal set of classifiers for a given dataset.

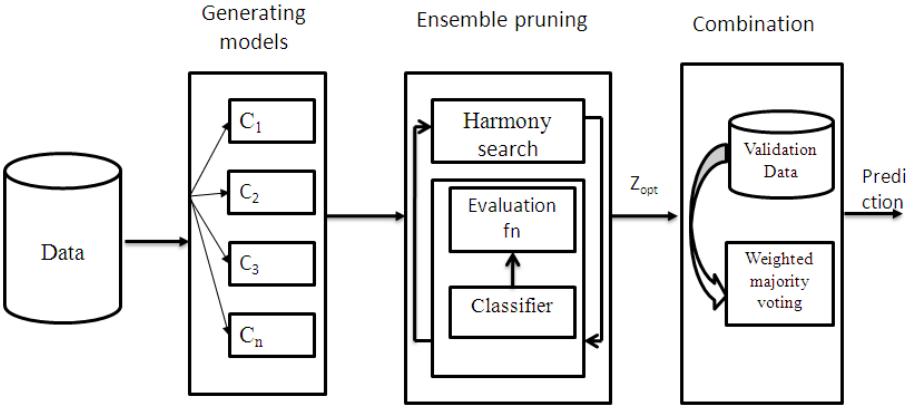


Fig. 1. Proposed model

The weak classifiers which don't contribute much to the final decision making process are eliminated at this stage. Ensemble combination is further done using weighted majority voting. The pseudo code for the proposed model HS_ENSEM is given in figure 2. The loop in line 2 trains the various classifiers once and the hypothesis is stored. Lines 14-17 constructs the harmony memory and generates binary solution vectors and calculates the fitness for each vector. The loop in line 20 creates a new solution vector based on the value of HMCR and PAR. The fitness of the new vector is calculated and it is compared with the values in harmony memory in line 28. If it is better than the worst harmony in memory it is replaced. After NI number of iterations the solution vector with the highest fitness is returned.

Algorithm HS_ENSEM
Input D: Dataset, Ω : Ensemble of Classifiers {C_1, ……., C_n}
Z_{opt} : Pruned ensemble set

1. Begin
2. For i = 1 to n
3. Train the classifiers with D and obtain hypothesis h
4. Endfor
5. Z_{opt} = harmonyensemble(Ω)
6. End
7. **Testing**
   a. Input an unlabeled instance X
   b. Evaluate X using Z_{opt}
   c. Obtain the composite hypothesis by weighted majority voting
   d. Choose the class with the maximum weighted votes.
8. **harmonyensemble(Ω)**
9. Begin
10. Initialize the parameters PAR, HMCR,NVAR,NI,BW,HMS
11. Int array X[];
12. Initialize X[]=0; t=0
13. Fitness=Accuracy of the ensemble
14. For i= 1 to HMS do
15. Generate random binary solutions and append it to HM
16. F[i] = Fitness value of the ith solution vector
17. End for
18. Worstfit = min(F[i])
19. While (t < NI)
20. For i=1 to NVAR
21. if (rand(0,1) < HMCR)
22. {
23. X[i] = HM[rand(0, HMS -1), i]
    a. if (rand(0,1) < PAR)
    b. flip the bit corresponding to X[i]
24. }
25. else
26. Randomly select the value of X[i] from population
27. end while
28. If fitness value of X[i] <= worstfit then
29. replace the vector in HM corresponding to worstfit with X[]
30. End if
31. End while
32. opt = max(F[i])
33. Z_{opt}= X[] corresponding to opt
34. Return Z_{opt}
35. End

Fig. 2. Pseudocode of the proposed model

The worst case time complexity of the algorithm is $O(n^2m)$ where 'n' is the number of classifiers and 'm' is the size of the data set. It has the same complexity as that of Genetic algorithm based pruning techniques but shows a better performance in terms of accuracy. The proposed method outperforms Hill climbing approach in terms of time which has a complexity of $O(n^2\ g(n;m))$, where $g(n;m)$ concerns the complexity of the evaluation process, which is linear with respect to 'm' and ranges from constant to quadratic with respect to 'n'.

5 Experimental Analysis

Six different algorithms implemented in Weka[20] were used to generate the independent heterogeneous classifiers for the experiments. The datasets considered are from the UCI machine learning repository [21]. The model was run for 20 iterations. Each harmony was evaluated with the accuracy as the objective function and the best harmonies were carried over to the next iteration. Larger harmony memory did not have an effect on performance.

The proposed method's accuracy was also compared to two commonly used ensemble techniques such as AdaBoost and Bagging and it is seen that it performs better in both instances.

In terms of computation performance and robustness, harmony search based approaches are computationally inexpensive themselves, because the algorithm comprises a very simple concept, and the implementation is also straightforward. The actual run time of the entire classifier ensemble process is then determined by two main factors, the max number of iterations, and the efficiency of the accuracy evaluation method.

We ran various experiments to test our approach. The following learning algorithms were used for the experiments: C4.5, PART, OneR, Naïve bayes, RBF Networks and Logistic Regression. Six data sets from the UCI repository were used. Each experiment was done using 10 fold cross validation for accuracy evaluation.

Table 3. Accuracy of the existing ensemble methods and our model

Ensemble methods	*Heart*	*Iris*	*Tictac*	*Kr-vs-kp*	*Pendigits*	*Satimage*
Bagging	81.25	94.00	92.068	99.123	99.2358	89.65
AdaBoost	**85.00**	95.33	72.547	93.836	20.4239	43.079
DECORATE	67.50	95.33	93.632	99.311	**99.7817**	89.113
Random forest	71.25	95.33	93.006	98.811	98.09	89.81
Random Subspace	58.28	98.657	88.309	99.780	99.09	88.958
Stacking	76.25	33.333	97.172	52.221	20.4076	23.9502
Our Model	76.88	**98.6577**	**97.178**	**99.780**	99.554	**92.140**

Tests were done using various state of the art ensemble combination techniques. The various techniques that we have compared our work with includes Bagging, AdaBoost, DECORATE, Random forest, Random Subspace and Stacking.The results are shown in table 3.

The accuracy of our proposed model is compared to that of EVEN [22] which is a genetic algorithm based ensemble pruning technique. Table 4 shows the results of comparing with their approach. We have used two data sets Pendigits and Satimage used in [22] and the results have been found to be encouraging.

The use of harmony memory in HS offers a major advantage over that of techniques like genetic algorithms, as it maintains a record of the historical data processed by previous iterations. All elements of the memory together contribute to the new harmony, while changes in genetic populations result in the destruction of previous knowledge of the problem. Harmony memory considering rate and pitch adjustment rate also help greatly in escaping from the local best solution.

Table 4. Comparison of our model with EVEN using Genetic algorithm

	Pendigits	*Satimage*
EVEN	99.21	89.94
Our Model	**99.554**	**92.17**

Fig 3 and Fig 4 shows the effect of HMCR and PAR on the accuracy of the model. It is seen that the best performance is for HMCR=0.9 and PAR=0.1.

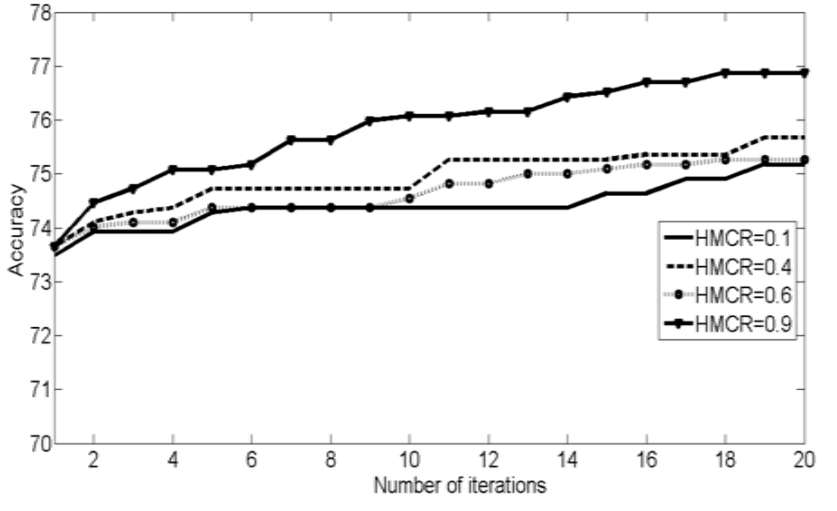


Fig. 3. Effect of HMCR on accuracy

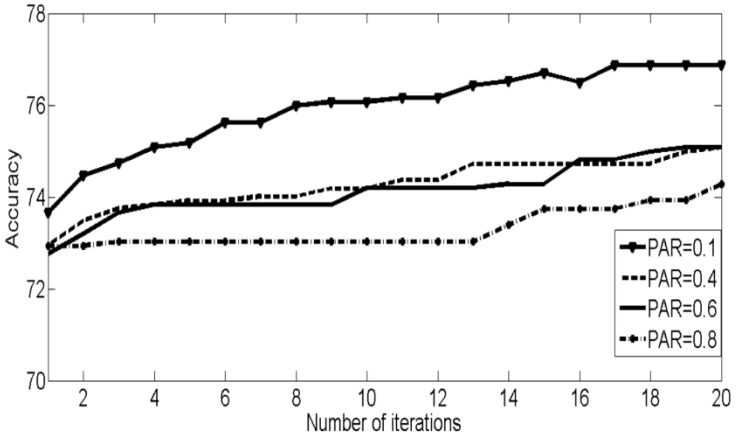


Fig. 4. Effect of PAR on accuracy

6 Conclusion and Future Work

A music inspired algorithm, harmony search is used to identify the most optimal ensemble of classifiers. The ensemble build using this model is found to be superior to the various state of the art techniques available today. It is also seen that ensemble learning provides a better performance than individual classifiers. Even though the computational cost of the proposed model is almost same as that of genetic algorithm based pruning, our model performs better in terms of accuracy.

As a possible extension of our work we plan to evaluate our method with the various evolutionary algorithms available and apply it for ensemble pruning in the domain of malware detection.

Acknowledgement. This work is a part of the Collaborative Directed basic Research on smart and secure environment project sponsored by NTRO, India.

References

1. Kuncheva, L.I.: Combining pattern classifiers: methods and algorithms. John Wiley & Sons Inc. (2004)
2. Breiman, L.: Bagging predictors. Mach. Learn. 24(2), 123–140 (1996)
3. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. IEEE Transactionson Pattern Analysis and Machine Intelligence 28(10), 1619–1630 (2006)
4. Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: A comparison of decision tree ensemble creation techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(1), 173–180 (2007)

5. Freud, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: Saitta, L. (ed.) Machine Learning: Proceedings of the Thirteenth International Conference (ICML 1996), pp. 148–156. Morgan Kaufmann, San Francisco (1996)
6. Wolpert, D.H.: Stacked Generalization. Neural Networks 5(2), 241–259
7. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
8. Prodromidis, A.L., Stolfo, S.J., Chan, P.K.: Effective and efficient pruning of metaclassifiers in a distributed Data Mining system. Technical report CUCS-017-99, Columbia Univ. (1999)
9. Caruana, R., Niculescu-Mizil, A., Crew, G., Ksikes, A.: Ensemble selection from libraries of models. In: Twenty-First International Conference on Machine Learning, Banff, Alberta, Canada, July 04-08 (2004)
10. Hu, Q., Yu, D., Xie, Z., Li, X.: EROS: Ensemble rough subspaces. Pattern Recognition 40, 3728–3739 (2007)
11. Margineantu, D., Dietterich, T.: Pruning adaptive boosting. In: Proc. Fourteenth Intl. Conf. Machine Learning, pp. 211–218 (1997)
12. Rokach, L., Arbel, R., Maimon, O.: Selective Voting - Getting More For Less in Sensor Fusion. International Journal of Pattern Recognition and Artificial Intelligence 20(3), 329–350 (2006)
13. Prodromidis, A.L., Stolfo, S.J.: Cost Complexity-Based Pruning of Ensemble Classifiers. Knowl. Inf. Syst. 3(4), 449–469 (2001)
14. Zhang, Y., Burer, S., Street, W.N.: Ensemble pruning via semi-definite programming. Journal of Machine Learning Research 7, 1315–1338 (2006)
15. Zhou, Z.H., Tang, W.: Selective Ensemble of Decision Trees. In: Wang, G., et al. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 476–483. Springer, Heidelberg (2003)
16. Tsoumakas, G., Partalas, I., Vlahavas, I.: An Ensemble Pruning Primer. In: Okun, O., Valentini, G. (eds.) SUEMA 2009. SCI, vol. 245, pp. 1–13. Springer, Heidelberg (2009)
17. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: Harmony search. Simulation 76(2), 60–68 (2001)
18. Geem, Z.W. (ed.): Music-Inspired Harmony Search Algorithm: Theory and Applications. SCI. Springer, New York (2009)
19. Lee, K.S., Geem, Z.W.: A new metaheuristic algorithm for continuous engineering optimization: Harmony search theory and practice. Comput. Methods Appl. Mech. Eng. 194(36-38), 3902–3933 (2004)
20. 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)
21. http://repository.seasr.org/Datasets/UCI/arff/
22. Sylvester, J., Chawla, N.: Evolutionary Ensembles: Combining Learning Agents using Genetic Algorithms. In: Proc. of AAAI Workshop on Multi-agent Systems, pp. 46–51 (2005)

A First Study on Decomposition Strategies with Data with Class Noise Using Decision Trees

José A. Sáez[1], Mikel Galar[2], Julián Luengo[3], and Francisco Herrera[1]

[1] Department of Computer Science and Artificial Intelligence of the University of Granada, CITIC-UGR, Granada, Spain, 18071
{smja,herrera}@decsai.ugr.es
[2] Department of Automática y Computación, Universidad Pública de Navarra, Pamplona, Spain, 31006
mikel.galar@unavarra.es
[3] Department of Civil Engineering, LSI, University of Burgos, Burgos, Spain, 09006
jluengo@ubu.es

Abstract. Noise is a common problem that produces negative consequences in classification problems. When a problem has more than two classes, that is, a multi-class problem, an interesting approach to deal with noise is to decompose the problem into several binary subproblems, reducing the complexity and consequently dividing the effects caused by noise into each of these subproblems. This contribution analyzes the use of decomposition strategies, and more specifically the One-vs-One scheme, to deal with multi-class datasets with class noise. In order to accomplish this, the performance of the decision trees built by C4.5, with and without decomposition, are studied. The results obtained show that the use of the One-vs-One strategy significantly improves the performance of C4.5 when dealing with noisy data.

Keywords: Noisy Data, Class Noise, One-vs-One, Decomposition Strategies, Ensembles, Classification.

1 Introduction

Any classification problem [1] consists of m training patterns, characterized by n attributes A_i, $i = 1, \ldots, n$, which are either numerical or nominal, being $\mathbb{D}_i$ their corresponding domains. Thus, an example $\mathbf{x}$ is represented as an n-dimensional attribute vector

$$\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{D} = \mathbb{D}_1 \times \cdots \times \mathbb{D}_n \ .$$

Each example is labeled with one of M possible classes $\mathbb{L} = \{\lambda_1, \ldots, \lambda_M\}$. Many current real-world classification problems, such as cancer classification [2] or the recognition of fingerprints [3], are characterized by having more than two classes, that is $M > 2$. These problems are formally known as multi-class classification problems.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 25–35, 2012.

Classification algorithms aim to extract the implicit knowledge from previously known instances of these problems by creating a model, called a classifier, that generalizes the peculiarities of the set of labeled examples and is capable of predicting the class for previously unobserved examples. Hence, the classification accuracy of a classifier is directly influenced by the quality of the training data used to build the model. Data quality depends on several components [4], for instance, the source of that data and the input of the data, inherently subject to errors, among others. Real-world datasets rarely avoid this type of errors and they usually contain corrupted data that may hinder the interpretations, decisions and therefore, the models created from the data.

Generally, the more classes in a problem, the more complex the decision boundaries are. Moreover, the presence of noise in such problems adds an extra complexity. Traditionally, decomposing a multi-class problem into several binary, easier to solve subproblems, has been related to obtaining a good performance when data are affected by noise –although this issue has not been explicitly addressed yet. In such a way, the complexity of the original problem is decreased, and as a consequence, noisy instances are divided into each subproblem, which also decreases the noise effect on the final performance of the classifier. These techniques are called binary decomposition strategies [8]. The *One-vs-One* (OVO) [9] and *One-vs-All* (OVA) [10] schemes are the most studied in the literature. OVO is based on dividing the problem into as many binary problems as possible combinations between pairs of classes, while OVA learns a classifier for each class. Generally, OVO outstands over OVA as reflected in the literature [11], [12], [13].

In this work our aim is to analyze the suitability of the OVO binary decomposition strategy with training data suffering from class noise. We will study the differences between OVO and non-OVO (baseline) classifiers built by C4.5 [6] through an analysis of their accuracy, which we will also contrast using the proper statistical tests as recommended in the specialized literature [14]. Notice that C4.5 is capable of handling multiple classes inherently; hence, we will be able to compare the OVO scheme with their baseline performances. In order to validate our hypothesis and to extract meaningful conclusions, we will prepare an experimental framework considering 21 real-world datasets. Four different levels of noise are introduced in the class labels in the training partitions: 5%, 10%, 15% and 20%. Thus, 84 new synthetic datasets with class noise will be created. The test sets will remain unchanged in order to check which strategy, OVO or baseline, performs better in the presence of noisy data.

The rest of this contribution is organized as follows. Section 2 presents an introduction to classification with noisy data. Section 3 is devoted to the motivations for the use of binary decomposition strategies in multi-class classification problems, recalling the OVO decomposition scheme. Next, Section 4 describes the experimental framework. Section 5 includes the analysis of the experimental results obtained by the classifiers with and without the use of the OVO decomposition scheme. Finally, in Section 6 we present our concluding remarks.

2 Classification with Noisy Data

Real-world data is never perfect and often suffers from corruptions that harm the interpretations of the data, the models created and the decisions made. In classification, noise can negatively affect the system performance in terms of classification accuracy, building time, size and interpretability of the classifier built [5].

The quality of any dataset is determined by a large number of components as described in [4]. Some of these are the source of the data and the input of the data, which are inherently subject to error.

Class labels and attributes are two information sources which can influence the quality of a classification dataset. The quality of the class labels represents whether the class of each instance is correctly assigned; and the quality of the attributes indicates how well the attributes characterize instances for classification purposes. Based on these two information sources, we can distinguish two types of noise in a given dataset [5]:

1. *Class noise*. These errors occur when an instance belongs to the incorrect class. Class noise can be attributed to several causes, including subjectivity during the labeling process, data entry errors, or inadequacy of the information used to label each object. There are two possible types of class noise:
   - Contradictory examples: the same examples appear more than once and are labeled with different classes.
   - Misclassifications: instances are labeled with the wrong classes [18].
2. *Attribute noise*. It is used to refer to corruptions in the values of one or more attribute of instances in the dataset. Examples of attribute noise include: erroneous attribute values, missing or unknown attribute values, and incomplete attributes or "do not care" values.

The two most common approaches to noisy data in the literature are:

- *Robust learners*. They are characterized by being less influenced by noisy data. An example of a robust learner is the C4.5 algorithm [6]. C4.5 uses pruning strategies to reduce the chances of trees being built with noise in the training data. However, when the noise level becomes relatively high, even a robust learner may obtain a poor performance.
- *Noise preprocessing techniques*. They are classifier-independent and try to remove the negative impact of noise in the datasets prior to creating a model over the original data. Among these techniques, the most well-known methods are noise filtering ones [7].

In this contribution, we focus on class noise because it is very common in real-world data [5], [18]. These errors can be produced in situations where different classes have similar symptoms, as generally happens on the class boundaries. We compare the performance of the C4.5 robust learner considering or not the use of decomposition. We want to verify that the effect of class noise on the accuracy of the decision trees created by C4.5 is lower considering the use of OVO, even if this classification algorithm is a robust learner.

3 Addressing Multi-class Classification by Decomposition

Classification tasks involving more than two categories or classes, commonly known as multi-class classification problems, are frequent in real-world problems. Multi-class problems are more general than the special case considering only two classes (binary classification problems).

A multi-class classification problem is intrinsically more complex than a binary one since the generated classifier must be able to separate the data into a higher number of categories, which increases the chances of incorrect classifications (in a two-class balanced problem, the probability of a correct random classification is $1/2$, whereas in a multi-class problem it is $1/M$).

In order to reduce the complexity of the original problems and/or to be able to use binary classification techniques to solve multi-class classification problems, in the literature two approaches have been adopted:

- Adaptation of the internal operations of the learning algorithm.
- Decomposition of the multi-class problem into a set of easier to solve two-class problems.

The extension of a binary learning algorithm to a multi-class version may be very difficult to perform in many cases [15]. Therefore, it is more common to use the alternative which decomposes the multi-class problem into binary subproblems, a strategy called decomposition.

3.1 Decomposition Strategies for Multi-class Problems

Several motivations for the use of binary decomposition strategies in multi-class classification problems can be found in the literature [11], [12]. For example, in [12], the reduction of the complexity involved in the classes' separation when using a decomposition approach was shown. Also in [16], the authors point out the advantages of the use of binary decompositions when the classification errors for different classes have distinct costs. This way, the binary predictors generated may impose preferences for some of the classes. Decomposition also opens up new possibilities for the use of parallel processing, since the binary subproblems are independent and can be solved with different processors.

Dividing a problem into several new problems which are then independently solved implies the need for a second phase where the outputs of each problem have to be aggregated. Therefore, decomposition includes two steps:

1. *Problem division.* In this phase, the problem is decomposed into several binary subproblems which are solved by independent binary classifiers, called base classifiers [12]. Different decomposition strategies can be found in the literature [8], the most common strategies are OVO [9] and OVA [10], as discussed above.
2. *Combination of the outputs* [11]. In this phase, the different outputs of the binary classifiers are aggregated in order to output the final class prediction. The simplest method is a voting strategy where each classifier gives a vote,

and the final prediction is given by the class achieving the largest amount of votes.

In this contribution, we consider the OVO decomposition strategy due to its several advantages shown in the literature [11], [12]:

- OVO creates simpler borders between classes than OVA. This is one of the main advantages of OVO that we want to exploit when training with noisy data. In such a way, the noise's corruptions in these regions will be less notable and the classifiers will be less influenced. Moreover, as OVO only distinguishes between two classes, if the noisy examples do not belong to one of the two classes that have been learned to be distinguished by a concrete classifier, this classifier will not be affected by noise and its predictions will not be altered.
- OVO generally obtains a higher classification accuracy and a shorter training time than OVA because it creates easier and smaller problems.
- OVO has less tendency to create imbalanced datasets which can be counterproductive [13].

In [11], an exhaustive study comparing different methods to combine the outputs of the base classifiers in the OVO and OVA strategies has been developed. However, the most used combination, also used in our experiments, is the voting strategy already mentioned.

3.2 One-vs-One Decomposition Scheme

The OVO decomposition strategy is based on dividing a classification problem with M classes, $\mathbb{L} = \{\lambda_1, \ldots, \lambda_M\}$, into $M(M-1)/2$ binary problems. Each new subproblem only considers the examples of the training data corresponding to a different pair of classes (λ_i, λ_j), with $i < j$.

In the learning phase, a binary classifier is created for each problem, which is capable of distinguishing between a different pair of classes. In the validation phase, an example is presented to each one of the binary classifiers. This way, each classifier discriminating between classes λ_i and λ_j provides a confidence degree $r_{ij} \in [0, 1]$ in favor of the former class, and another confidence degree in favor of the latter $r_{ji} \in [0, 1]$ (if the classifier does not provide the latter, it is computed by $r_{ji} = 1 - r_{ij}$). These outputs are represented by a score matrix R:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix}. \tag{1}$$

The final output of the system is derived from the score matrix by different aggregation models. As we have previously mentioned, the voting strategy is the simplest method:

$$Class = \arg\max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} s_{ij} \tag{2}$$

where s_{ij} is 1 if $r_{ij} > r_{ji}$ and 0 otherwise. This strategy has shown a competitive behavior with different classifiers [11] obtaining similar results in comparison with more complex strategies.

Although the number of classifiers is of M^2 order, as each classifier is only trained with examples from two classes, the required time is distributed, and hence is usually low. However, there is also a drawback: when a new example is submitted to all the classifiers, some of them may not have seen a similar instance before, so their output would not be significant; these cases are called non-competent examples [17]. In any case, OVO aggregations usually suppose that the base classifiers will be correctly predicted if the new pattern is one of the considered pairs of classes, and therefore, considering a voting strategy, the class with the largest number of votes will be the correct class.

4 Experimental Framework

In this section, we present the details of the experimentation developed in this contribution. First, in Subsection 4.1, we describe the base datasets of our experimentation. Then, we show how to build up the noisy datasets from the base ones in Subsection 4.2. Finally, the methodology for the analysis of the results is explained in Subsection 4.3.

4.1 Base Datasets

The experimentation is based on 21 real-world multi-class classification problems from the UCI repository[1]. Table 1 shows the datasets sorted by the number of classes (#Cla). Moreover, for each dataset, the number of instances (#Ins) and the number of attributes (#Att) along with the number of real, integer and nominal attributes (R/I/N) are presented. Some of the largest data-sets (nursery, page-blocks, penbased, satimage, shuttle and led7digit) were stratified at 10% in order to reduce the computational time required for training. For datasets containing missing values (automobile and dermatology), these instances with missing values were removed from the dataset before the partitioning.

4.2 Inducing Noise in Datasets

The initial amount of noise present in the previous datasets is unknown so we cannot make any assumption about it. We need to control in some way the amount of noise in each dataset. This will help us to check how a higher or a lower amount of noise affects the models obtained by the classification algorithms. For these reasons, we systematically and independently add noise to each dataset, as proposed in [5].

In order to introduce a level of class noise of $x\%$ in a dataset, we use a pairwise scheme as indicated in [18]: given a pair of classes (X, Y), with X the majority

[1] http://archive.ics.uci.edu/ml/datasets.html

Table 1. Summary description for classification datasets

Dataset	#Cla	#Ins	#Att (R/I/N)	Dataset	#Cla	#Ins	#Att (R/I/N)
balance	3	625	4 (4/0/0)	glass	7	214	9 (9/0/0)
contraceptive	3	1 473	9 (0/9/0)	satimage	7	643	36 (0/36/0)
iris	3	150	4 (4/0/0)	segment	7	2 310	19 (19/0/0)
splice	3	319	60 (0/0/60)	shuttle	7	2 175	9 (0/9/0)
thyroid	3	720	21 (6/15/0)	zoo	7	101	16 (0/0/16)
wine	3	178	13 (13/0/0)	ecoli	8	336	7 (7/0/0)
nursery	5	1 269	8 (0/0/8)	led7digit	10	500	7 (7/0/0)
page-blocks	5	547	10 (4/6/0)	penbased	10	1 099	16 (0/16/0)
automobile	6	150	25 (15/0/10)	yeast	10	1 484	8 (8/0/0)
dermatology	6	358	34 (0/34/0)	vowel	11	990	13 (10/3/0)
flare	6	1 066	11 (0/0/11)				

class and Y the second majority class, and a noise level $x\%$, an instance with the label X has a probability of $x\%$ to be incorrectly labeled as Y. As indicated in [5], this scheme is appropriate because it is more likely that only certain types of classes are mislabeled.

In order to create a noisy dataset from the original one, the noise is consistently introduced into the training partitions as follows:

1. A level of noise $x\%$ is introduced into a copy of the full original dataset.
2. Both datasets, the original one and the noisy copy, are partitioned into 5 equivalent folds having the same examples per fold.
3. We use a cross-validation scheme for new noisy datasets, building the training partitions with the noisy copy, and the test partitions with the original dataset.

The accuracy estimation of each classifier in a dataset is obtained by means of 5 runs of a stratified 5-fold cross-validation. The dataset is divided into 5 partition sets with equal numbers of examples and maintaining the proportion between classes in each fold. Each partition set is used as a test set for the model learned from the four remaining partitions. This procedure is repeated 5 times. We use 5 partitions as if each partition has a large number of examples, the noise effects will be more notable, facilitating their analysis.

Introducing noise into training partitions makes it possible to observe how noise affects the test accuracy of the classifiers when training with noisy data. The accuracy of the model built over the original training set without additional noise can act as a reference value. In this way, we can observe how the accuracy of the models built with noisy training sets is more or less affected with respect to this value by the noise effect.

From the 21 base datasets from the UCI repository we have created a large collection of new noisy datasets. We have studied the levels of noise: $x = 5\%$, $x = 10\%$, $x = 15\%$ and $x = 20\%$. Therefore, 84 datasets with class noise were created.

4.3 Analysis Methodology

The main aim of this contribution is to check whether the use of the OVO binary decomposition strategy improves the classification performance in multi-class datasets affected by class noise. We will study the advantages provided by this strategy against not using it in this framework. For this reason, we consider the C4.5 algorithm which can deal with multi-class problems directly, and we use the OVO scheme in order to find whether there are improvements with respect to the original algorithm that does not use it.

In order to be able to make this comparison, we use the mean accuracy provided by C4.5 over the test sets for each level of induced noise, defined as its performance averaged across all classification problems. Along with the test accuracy, we use the Wilcoxon signed ranks statistical test [14]. This is a non-parametric pairwise test that aims to detect significant differences between two sample means; that is, the behavior of the two implicated algorithms in the comparison. Statistical analysis needs to be carried out in order to find significant differences among the results obtained by the studied methods. Accordingly, we do not only consider the mean accuracy, but we also take into account the statistical differences. Therefore, our conclusions are not only based on averaged mean results. For each level of noise, we compare C4.5 using OVO versus C4.5 trained with the complete dataset with the Wilcoxon test and we obtain the p-values associated with these comparisons.

5 Analysis of the One-vs-One Decomposition Strategy with Data with Class Noise

In this section we analyze the performance of C4.5 using the OVO decomposition with respect to its baseline results when dealing with data with class noise. Table 2 shows the test accuracy results of C4.5 in each single dataset (with and without OVO) and also the mean test accuracy as an indicator at each noise level. Table 3 shows the associated p-values of the Wilcoxon test between the OVO and non-OVO version at each noise level.

From these two tables of results we should stress several points:

- The good performance of C4.5 when using the OVO strategy must be highlighted, since the test accuracy increases with respect to the absence of decomposition.
- Although C4.5 is considered a robust learner tolerant to class noise, the binary decomposition into subproblems causes the algorithm to achieve better accuracy rates than baseline at all noise levels.
- Moreover, as shown in Table 3, we can observe from the associated p-values that there are significant differences in the results of C4.5 when using OVO with respect to the baseline results, in favor of OVO. This occurs for all noise levels.

Table 2. Test accuracy results on each single dataset with class noise. The best results between the OVO and the non-OVO version for each noise level have been stressed in bold.

	C4.5 without decomposition					C4.5 with OVO				
Noise %	**0%**	**5%**	**10%**	**15%**	**20%**	**0%**	**5%**	**10%**	**15%**	**20%**
autos	76.7339	78.0645	71.0484	72.9435	76.6935	**80.5242**	**79.8589**	**77.3387**	**77.3185**	**77.3790**
balance	77.2800	77.4400	77.6000	76.8000	76.9600	**80.1600**	**80.3200**	**81.4400**	**79.6800**	**78.2400**
contraceptive	**52.6798**	52.3406	50.6415	51.2561	48.8083	51.7297	**53.1588**	**51.7290**	**52.0689**	**50.0989**
dermatology	92.4648	92.4687	91.6236	93.0203	91.3498	**95.5321**	**94.1354**	**93.2981**	**94.9804**	**93.2942**
ecoli	78.2836	**79.4776**	77.6822	**77.0896**	77.6997	**78.5777**	77.1071	**80.0702**	76.8130	**78.5821**
flare	**74.4803**	**74.4803**	**74.4803**	**74.4803**	**74.4803**	74.2947	74.2947	74.2947	74.2947	74.2947
glass	68.7265	63.5991	65.4707	62.2148	**64.0421**	**71.9048**	**71.0188**	**67.2979**	**64.0310**	63.5327
iris	**93.3333**	**93.3333**	**93.3333**	**94.0000**	**92.6667**	**93.3333**	**93.3333**	**93.3333**	93.3333	92.0000
led7digit	70.6000	70.4000	70.0000	70.8000	70.8000	**71.8000**	**71.6000**	**72.0000**	**72.2000**	**72.0000**
newthyroid	91.1628	91.6279	89.7674	88.3721	**91.6279**	**94.4186**	**94.4186**	**92.5581**	**90.6977**	90.6977
nursery	**89.0446**	**89.0446**	**89.0446**	**89.0446**	**89.0446**	88.8907	88.9676	88.8138	88.7369	88.7369
page-blocks	97.0212	97.0030	96.8386	96.3269	96.4730	**97.1125**	**97.1126**	**97.1857**	**96.8384**	**97.0760**
penbased	96.1518	96.4701	96.2973	96.3609	96.2518	**97.0069**	**96.9341**	**96.6339**	**96.8158**	**96.8887**
satimage	85.5789	85.6410	84.2269	**83.6364**	82.9371	**87.0396**	**86.3869**	**84.6620**	**83.6364**	**85.0816**
segment	96.7100	96.5368	96.7100	96.1905	96.3636	**97.0130**	**96.7532**	**96.9697**	**96.7100**	**96.6234**
shuttle	99.5402	99.4943	99.5862	99.5402	98.9885	**99.7241**	**99.7241**	**99.6322**	**99.6782**	**99.1724**
splice	79.3105	78.9980	74.6230	74.2659	70.2282	**89.0179**	**85.2530**	**82.4454**	**82.7431**	**81.1756**
thyroid	**99.4861**	**99.4583**	**99.4444**	**99.3611**	**99.0556**	99.4722	99.4306	99.4167	99.2917	98.9167
vowel	79.4949	78.1818	77.0707	**78.6869**	76.8687	**79.7980**	**78.9899**	**78.7879**	78.0808	**77.5758**
wine	**94.9048**	90.9841	89.3016	92.6190	88.1746	92.1270	**92.6667**	**89.8413**	**93.7460**	**89.8889**
yeast	54.9181	55.7951	53.3697	54.0411	54.1792	**58.4239**	**58.2214**	**58.6257**	**58.2209**	**56.6043**
zoo	**94.0952**	**95.0476**	**95.0476**	**93.0952**	**93.1429**	93.0952	92.0952	92.0952	90.0952	90.1429
mean	83.7273	83.4494	82.4185	82.4612	82.1289	**85.0453**	**84.6264**	**84.0213**	**83.6369**	**83.0910**

Table 3. Test accuracy results and Wilcoxon's test p-values

Noise %	**0%**	**5%**	**10%**	**15%**	**20%**
p-value	**0.0129**	**0.0096**	**0.0017**	**0.0228**	**0.0262**

– There are several datasets with a noise level of 0%, that is, without additional noise introduced, such as *contraceptive* or *wine*, where the non-OVO version outperforms the OVO version. However, OVO perform better than non-OVO when we introduce noise into these datasets.

These results show the usefulness of decomposition strategies to deal with class noise. For this type of noise, the overall test accuracy of C4.5 using the OVO decomposition strategy is always better than that of its baseline classifier, at each level of induced noise. Also, as reflected by the Wilcoxon test p-values, a better and significant global behavior is shown when OVO is used. This clearly shows the better performance of C4.5 using this decomposition strategy in a noisy framework. This better behavior of the OVO scheme dealing with class noise can be attributed to two main causes:

– Decomposing the problem into several binary subproblems increases the separability of the classes, obtaining simpler and more regular borders between some pairs of classes, thereby facilitating the construction of the classifier. In such a way, more compact and general classifiers can be constructed.

– Collecting information from different models may provide a more robust method for classification in noisy environments than collecting information from a single model. Thus, if a noisy example does not belong to one of both classes involved in the training of a classifier, the classifier will not be affected by that noise, and its predictions will not be hindered.

6 Concluding Remarks

In this contribution we have analyzed the suitability of the OVO decomposition scheme when dealing with datasets with class noise in multi-class problems. We have created 84 datasets with class noise. We have tested the C4.5 algorithm over these datasets. This method has been compared in its original version, which directly address the multi-class problem, and considering the OVO decomposition strategy.

The test accuracy results have shown that C4.5 using OVO performs better when trained over noisy data than the baseline method. In addition, the statistical tests carried out have shown that these improvements using OVO are significant.

This better behavior of OVO with data with class noise can be attributed to two main causes: (1) decomposing the problem into several binary subproblems lead us to create simpler, easier to build, classifiers and (2) if a noisy example does not belong to one of both classes involved in the training of a classifier using OVO, the classifier will not be affected by that noise, and its predictions will not be hindered.

In future works, the consideration of other kinds and schemes of noise, e.g. the attribute noise; the incorporation of additional algorithms; or the comparison of the decomposition strategies with other preprocessing techniques to deal with noisy data can be interesting.

Acknowledgments. Supported by the Spanish Ministry of Science and Technology under Project TIN2011-28488 and also by Regional Project P10-TIC-6858. J. A. Sáez holds an FPU scholarship from the Spanish Ministry of Education and Science.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley, New York (2001)
2. Anand, A., Suganthan, P.N.: Multiclass cancer classification by support vector machines with class-wise optimized genes and probability estimates. Journal of Theoretical Biology 259(3), 533–540 (2009)
3. Hong, J.H., Min, J.K., Cho, U.K., Cho, S.B.: Fingerprint classification using one-vs-all support vector machines dynamically ordered with naïve bayes classifiers. Pattern Recognition 41(2), 662–671 (2008)

4. Wang, R.Y., Storey, V.C., Firth, C.P.: A Framework for Analysis of Data Quality Research. IEEE Transactions on Knowledge and Data Engineering 7(4), 623–640 (1995)
5. Zhu, X., Wu, X.: Class Noise vs. Attribute Noise: A Quantitative Study. Artificial Intelligence Review 22, 177–210 (2004)
6. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers, San Francisco (1993)
7. Brodley, C.E., Friedl, M.A.: Identifying Mislabeled Training Data. Journal of Artificial Intelligence Research 11, 131–167 (1999)
8. Lorena, A., de Carvalho, A., Gama, J.: A review on the combination of binary classifiers in multiclass problems. Artificial Intelligence Review 30, 19–37 (2008)
9. Knerr, S., Personnaz, L., Dreyfus, G.: Single-Layer Learning Revisited: A Stepwise Procedure for Building and Training a Neural Network. In: Fogelman Soulié, F., Hérault, J. (eds.) Neurocomputing: Algorithms, Architectures and Applications, pp. 41–50. Springer, Heidelberg (1990)
10. Anand, R., Mehrotra, K., Mohan, C.K., Ranka, S.: Efficient classification for multiclass problems using modular neural networks. IEEE Transactions on Neural Networks 6(1), 117–124 (1995)
11. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recognition 44(8), 1761–1776 (2011)
12. Furnkranz, J.: Round Robin Classification (2002)
13. Sun, Y., Wong, A. K. C., Kamel, M. S.: Classification of Imbalanced Data: a Review. International Journal of Pattern Recognition and Artificial Intelligence, 687–719 (2009)
14. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7, 1–30 (2006)
15. Passerini, A., Pontil, M., Frasconi, P.: New results on error correcting output codes of kernel machines. IEEE Transactions on Neural Networks, 45–54 (2004)
16. Pimenta, E., Gama, J.: A study on error correcting output codes. In: Portuguese Conference on Artificial Intelligence EPIA, pp. 218–223 (2005)
17. Fürnkranz, J., Hüllermeier, E., Vanderlooy, S.: Binary Decomposition Methods for Multipartite Ranking. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS, vol. 5781, pp. 359–374. Springer, Heidelberg (2009)
18. Zhu, X., Wu, X., Chen, Q.: Eliminating class noise in large datasets. In: Proceeding of the Twentieth International Conference on Machine Learning, pp. 920–927 (2003)

Combining the Advantages of Neural Networks and Decision Trees for Regression Problems in a Steel Temperature Prediction System

Miroslaw Kordos[1], Piotr Kania[1], Pawel Budzyna[1],
Marcin Blachnik[2], Tadeusz Wieczorek[2], and Slawomir Golak[2]

[1] University of Bielsko-Biala, Department of Mathematics and Computer Science,
Bielsko-Biala, Willowa 2, Poland
mkordos@ath.bielsko.pl
[2] Silesian University of Technology, Department of Management and Informatics,
Katowice, Krasinskiego 8, Poland
marcin.blachnik@polsl.pl

Abstract. Simple decision trees enable obtaining simple logical rules with a limited accuracy in regression tasks. Neural networks as highly non-linear systems can map much more complex shapes and thus can obtain higher prediction accuracy in regression problems, that is however at the cost of the poor comprehensibility of the decision process. We present a hybrid system which incorporates the features of both a regression tree and a neural network. This system allowed for achieving high prediction accuracy supported by comprehensive logical rules for the practical problem of temperature prediction in electric arc furnace at one of the steelwors.

Keywords: neural networks, decision tree, regression, logical rules.

1 Introduction

There are some differences between extracting logical rules for classification and for regression tasks. A good strategy in data mining for the classification task is to extract simplest crisp logical rules first, because they provide hyper-rectangular decision borders in the feature space and that kind of rules are easy to understand. In the regression tasks also simple logical rules are preferred, however the issue is more complex because of the continuous output space. Several approaches to that issue have been attempted so far, including simple decision trees, which really provide hyper-rectangular decision borders, but which are frequently not the most accurate solution. Also rule extraction from standard multilayer perceptrons was performed, which frequently provides quite a good approximation, but the extracted logical rules can be of very high complexity and difficult to understand by experts in a given domain. In order to address this issue we tried two approaches: to improve the accuracy of a decision tree and to improve the comprehensibility of rules extracted from a neural network. Although hybrid methods can obtain very good accuracy [1,2], the customers are not always happy with them, because of the level of knowledge required to understand the systems. Thus, each of the

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 36–45, 2012.

solutions was aimed to be used as the only one kind of prediction model, because this homogeneity is more preferred by our customers. The purpose of the methodology we present in this paper is to achieve both goals; high accuracy and low rule complexity as far as possible by using only a single neural network-based model.

This section outlines the problems associated with neural network-based rule extraction for regression tasks and the problems connected with temperature control in electric arc steel-making. In the Methodology section we present our system and in the Experimental Evaluation section we compare it to some other systems on the two metallurgical datasets that can be obtained from [3] and on two datasets from the UCI repository [4,5].

1.1 Neural Network-Based Rule Extraction for Regression Problems

In our paper presented at the previous HAIS conference [6] we considered optimization of regression tree parameters. Although the logical rules were very clear and allowed for a better understanding of the process, we were able to obtain much higher accuracy for the same task using an ensemble of neural networks with evolutionary optimization [8]. Thus, for the practical purposes, where the system had to be used in production environment in the metallurgical industry, we decided to use a committee of several models. In our current approach we want to simplify this committee by introducing a new regression and rule extraction model based on a specially designed and specially trained neural network.

Most of the research concerns rule extraction for classification tasks, which is much easier. So far only a few approaches to rule extraction from neural networks for regression tasks have been described in the literature.

Setiono et. al. [9] and Wang et. al. [10] proposed an approach where each rule in the extracted rule set corresponds to a subregion of the input space. The nonlinear activation function (hyperbolic tangent or logistic sigmoid) of each hidden neuron is approximated locally by a three-piece linear function based on least squares estimation and then the regression rules are generated. Although in our experiments the accuracy of the rules obtained from that approach was only little below the accuracy of the underlying neural network, the complexity of the rules, when applied to our data was too high to allow understanding them quickly (in the time range below 1 minute, which is the time, the electric arc operator has to make decisions about how to further continue the process).

Kazumi [11] proposed a method where each regression rule is expressed as a pair of a logical formula on the conditional part over nominal variables and a polynomial equation on the action part over numeric variables. The proposed extraction method first generates one such regression rule for each training sample, then utilizes the k-means algorithm to generate a much smaller set of rules having more general conditions, where the number of distinct polynomial equations is determined through cross-validation. Finally, this method invokes decision tree induction to form logical formulae of nominal conditions as conditional parts of the final regression rules.

Markowska and Mularczyk [12] proposed o methodology based on two hierarchical evolutionary algorithms with multiobjective Pareto optimization. The lower level algorithm searches for rules that are optimized by the upper level algorithm. The conclusion

of the rule takes the form of a tree, where non-terminal nodes contain functions and operators and leaves contain identifiers of attributes and numeric constants.

1.2 Temperature Control in the Electric Arc Furnace

In the electric arc furnace the steel scrap is melted using the electric arc to generate most of the heat. Additional heat is obtained from gas that is inserted and burnt in the furnace. The optimal temperature of the melted steel that is to be tapped out from the furnace is about 1900K, however it must be kept at proper temperature enough long so that all the solid metal gets melted. If the heating lasts too long, unnecessary time and energy is wasted and additional wear of the furnace is caused. Modern EAFs have the melt times of 30 minutes, older ones up to one hour.

The temperature is measured a few times during every melt by special lances with thermocouple that are inserted into the liquid steel. Every measurement takes about one minute and in this time the arc has to be turn off and the process suspended. Waste of time and energy for three or even more measurements is thus quite significant. There are many problems with the continuous measurement of the steel temperature. The temperatures are very high and the radiant heat and the electro-magnetic radiation from the furnace creates major problems for the measuring equipment.

Therefore there was a need to build a temperature prediction system that would allow us to limit the number of temperature measurements and thus shorten the EAF process to make it more economical. We previously built a system based on a single regression model [6], as a part of the whole intelligent system for steel production optimization [14]. However, as our recent experiments showed, a significant improvement can be obtained with a committee of regression models. But on the other hand using complex committees of hybrid regression models made the extraction of simple comprehensive rules unrealistic in practice.

2 Methodology

2.1 Data Preprocessing

The presence of outliers and wrong data may dramatically reduce generalization abilities and limit the convergence of training process of any learning algorithm. Proper data preprocessing is helpful not only to deal with outliers but also to achieve variable sensitivity of the model in different ranges of input variables. First we standardize the data before the training, according to the following formula:

$$x_{std} = \frac{x - \bar{x}}{\sigma} \quad \sigma = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(x - \bar{x})^2} \tag{1}$$

and then we transform the data be the hyperbolic tangent function

$$y2 = \frac{1 - exp\,(-\beta \cdot y + \theta)}{1 + exp\,(-\beta \cdot y + \theta)} \tag{2}$$

where y is the output value before the transformation and $y2$ - after. In practical problems, it is frequently desired to obtain a model with higher sensitivity in the intervals with more dense data (as it was in our case) or in other intervals of special interests. To address his issue, we transfer the data through a hyperbolic tangent function. The other advantage of the transformation is the automatic reduction of the outliers' influence on the model. Because if the complexity of our data and lot of error in particular value measurements, before the values were recorded into the database it is very difficult to properly apply known outliers removal methods, such as ENN or other. For that reason we do not do not reject the outliers [13], but rather reduce their influence on the final model, because it is frequently not clear whether a given value is already an outlier or wrong value or is still correct. The hyperbolic tangent transformation allows for a smooth reduction of the outliers, because no matter how big the value is, after the transformation it will never be grater than one or smaller than minus one. however, in the case of multimodal data distribution the data should be first divided into several single-mode distribution datasets.

2.2 Network Construction

The neural network is based on a 3-layer MLP architecture. There are two kind of networks used in parallel. At the beginning of the training neurons implement hyperbolic tangent transfer functions, then in the rule-based network the transfer functions are gradually changed to step functions, while in the classical network they remain unchanged. The rule-based network requires discrete input data. If the data is continuous, what is frequently the case in regression tasks, it must be discretized prior to the training.

We tried using equi-distance and equal-number bins during the discretization. However, it turns out, that the best results can be achieved if the continuous attributes are discretized taking into account not only the input but also the output value. That prevents grouping into one bin the attribute values which correspond to a quite different output values and prevents making a split in the points for which output value does not differ. For that reason we use a decision tree as described in [6] for which there is only one input - the attribute under consideration. The decision tree divides the input attribute into bins in such a way, which minimizes the variance of each node. The division is performed so long as the predefined stopping criteria are met: the variance of the output value in the node, minimum number of vectors inn the node and maximum tree depth (For more details see [6]). In practice, with the metallurgical datasets on average the number of continuous feature bins (corresponding to the number of the tree leaves) was 4 to 16.

In the network for rule extraction a separate input neuron for each discretized feature value is used. Thus the number of all input neurons equals the sum of all distinct values for all features. The input values are 1 if the feature has the value represented by this neuron and 0 otherwise.

In practice we use a separate network for each output value bin. One hidden neuron per each output bin is created at the beginning. The second hidden neuron per each output value bin is added, if the results with only one neuron are not satisfactory and then the weights of the first neuron are frozen and only the newly added neuron is trained. If the results are still unsatisfactory then the next hidden neuron is added. The number of

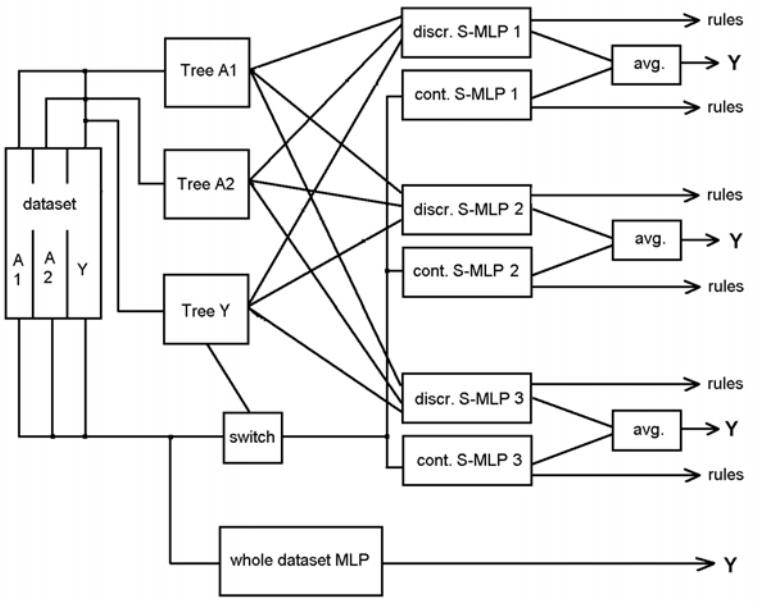


Fig. 1. The system used for temperature prediction in the EAF process

hidden neurons per one output bin should equal the number of data clusters that correspond to that output value bin. The first hidden neuron corresponds to the biggest cluster and thus generates the most general rules, while the second hidden neuron generates the rules for the second largest cluster and so on.

After the network is converted to step transfer function at the final part of the training, each hidden layer neuron performs M-of-N operation, where frequently N=M and thus the operation can be reduced to AND operation. The output neuron combines the partial rules given by hidden neurons for a given output value bin (OR operation). The bias and weights of output neurons are constant; bias = 0.5 each weight = 1, what implements the OR operator.

2.3 Network Training

Only one network can be used for the whole training. That network would consist of 10 output neurons (if we assume, the output value is discretized into 10 bins), each being responsible for a different bin of the output value. However, it is much easier to train 10 separate networks, where each of them has only one output neuron and thus the regression problem can be reduced to 10 classification problems with two classes only (this bin and the rest). In the dataset we replace the value of the output with one of it is within the current bin and with zero otherwise (in a similar way as the input values).

We begin the training with hyperbolic tangent transfer functions at the hidden and output layer neurons. In the final stage of the training the transfer functions are gradually converted to step functions in order to be able to extract simple logical rules from the network. For that purpose a regularization tern t is added to the error function to gradually force the network weights to take the values of -1, 0 or 1.

$$t = k_w * sum((w-1)(w+1)w) + k_b * sum((b-x+0.5)(b-x+1.5)...(b+x-0.5)(b+x+0.5)) \quad (3)$$

where the first part of the equation regularizes the weight values and the second part the biases and x is the number of attributes in the discretized dataset.

From certain point of the training we begin gradually increasing the transfer function slopes as long until we practically obtain step transfer functions, which enables simple extraction of logical rules.

In parallel we use another ("classical") network, which is trained on continuous values and for which we do not increase the slopes of transfer functions. That network is trained only on those vectors, which have the output value within the current bin. From this network we extract logical rules in a method similar to that proposed by Setiono [9]. However, because we add the regularization term to the error function proportional to the sum of the weight squares and we use only vectors from one output bin, we can significantly simplify the system of equations used to extract the logical rules by assuming that all the neurons operates only within the linear part of the hyperbolic transfer function (-1,1). That can be achieved also, because the output value is 10 times less differentiated within one bin than within the whole range.

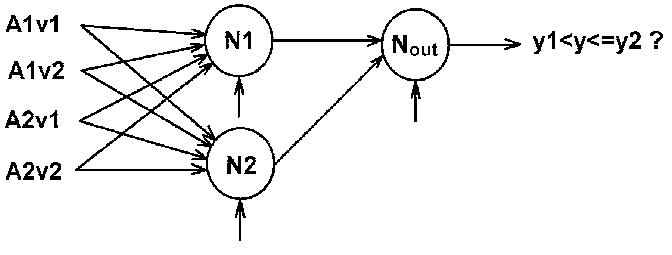


Fig. 2. The single discrete network for rule extraction (for one output bin)

In this way we obtain two sets of neural networks with two different sets of rules. The rules generated by the first set are simpler and frequently are sufficient for the expert. If the experts wants to have more accurate rules, they can use the second model (which generates more complex rules, but still much simpler as the approach presented in [9]). In practice a good combination is to provide the expert with two sets of rules, the two values predicted by the two networks and a final value, which we suggest to use

$$y = (y_1 + 2 * y_2)/3 \quad (4)$$

If several consecutive values of one feature have the same weights assigned at one neuron, then the rule can easily be simplified by merging all the values into one bin.

2.4 Network Test

At the test phase, the test vector is given as an input to all the 10 logical networks and it is assigned to that output bin, which network's output neuron sets its signal to 1. The signals of all other networks' output neurons should produce the value of -1 in response to that vector. However, that is not always the case and sometimes may happen that the networks responses don't allow us to unambiguously assign the vector to one output bin. In this cases we use a standard MLP network trained on the whole dataset. That network predicts some output value of the vector and we assign it to the appropriate bin according to that prediction.

3 Experimental Evaluation

3.1 Experimental Methodology

In this section we compare our method to a regression tree, to a forest of regression trees [6], an MLP network, a hierarchical committee of MLP networks and to the MLP-based method proposed by Setiono [9].

The single MLP neural network had the structure $n - n - 1$, where n is the number of attributes and was trained with the VSS algorithm [16]. In the hierarchical MLP Committee, the whole dataset was split into several clusters, as in the Mixture of Expert approach, however there was a hierarchy of clusters, where the clusters of higher levels contained the clusters of lower levels. Several neural networks were created and trained for each cluster, using bagging to select the vectors. Then an evolutionary algorithm was used to decide how a final decision of the committee must be evaluated based on the test vector properties and on particular neural network responses [7]. The last compared method was a method for extracting logical rules form an MLP network as described in [9], where the sigmoid transfer functions were approximated by a three straight lines and depending on the current point of the sigmoid a proper linear approximation was used for rule generation. The decision tree and forest were constructed as described in our paper presented at the previous HAIS conference [6].

We created software that implements all the methods in C#, Java and Delphi languages and made it available together with the datasets used in the experiments from our web page at [3]. All the methods were run in a 10-fold crossvalidation.

3.2 Datasets

The datasets we used are: two datasets depicting the metallurgical problem; one of them refers to the temperature prediction in the EAF process as described in the introduction and one refers to predicting the amount of carbon to be added in the steel refinement process. The next two datasets are the Crime and Concrete datasets from the UCI Machine Learning Repository.

Concrete Compressive Strength. There are 8 input attributes (variables) in the dataset reflecting the amount of particular substances in the concrete mixture, such as cement, slag, water, etc. The task is to predict the concrete compressive strength. There are

1030 instances in the dataset. The dataset is available from the UCI Machine Learning Repository [4].

Communities and Crime. There are 127 input attributes in the data set, describing various social, economical and criminal factors. The attribute to predict is per capita violent crime. After removing the instances with missing attributes, 121 instances were left. The dataset is available from the UCI Machine Learning Repository [5].

Steel-Carbon. The dataset comes from a real metallurgical process at the phase of refining the melted steel in a ladle arc furnace to achieve desired steel properties. The input variables represent various measured parameters, such as temperature, energy, amount of particular elements in the steel etc. The amount of carbon that should be added to the steel refinement process is the output variable. The data was standardized, only 12 attributes were left from the original dataset with over 100 attributes and the names of 12 input attributes were changed to $x_1 \ldots x_{12}$. There are 1440 instances in the dataset. The dataset is available from [3].

Steel-Temperature. The dataset comes from a real metallurgical process at the phase of melting the steel scrap in the electric arc furnace. The input variables represent various measured parameters, such as temperature, energy, amount of gases, etc. The temperature of the liquid steel is the output variable. The data was standardized, only 14 attributes were left from the original dataset with over 100 attributes and the names of 14 input attributes were changed to $x_1 \ldots x_{14}$. There are 7400 instances in the dataset. The dataset is available from [3].

3.3 Experimental Results

In regression and classification problems, there is no single model which is best for all datasets and models must be chosen for a given task. As it can be seen from table 1, usually the lowest MSE for the large metallurgical datasets is obtained with hierarchical MLP committee. For simple datasets, this model is too complex. However, the purpose of the work was to create a model fitted for rule extraction for large and complex datasets, especially in the application to temperature prediction in electric arc furnace. Thus, it cannot be stated that a model with a lower MSE error is better, because there are situation where a model that can provide simple rule is preferred over a model

Table 1. Experimental results; mean square error (MSE) and standard deviation in 10-fold cross-validation

dataset		single tree	tree forest	discrete Split-MLP	continues Split-MLP	single MLP (values)	single MLP (rules)	hierarchical MLP committee
Concrete	MSE	0.153	0.121	0.133	0.125	0.124	0.125	0.120
	std. dev.	0.020	0.018	0.015	0.020	0.016	0.017	0.014
Crime	MSE	0.35	0.30	0.32	0.28	0.28	0.30	0.28
	std. dev.	0.04	0.04	0.03	0.03	0.03	0.03	0.03
Steel-C	MSE	0.140	0.122	0.120	0.115	0.118	0.128	0.110
	std. dev.	0.017	0.015	0.018	0.014	0.015	0.017	0.011
Steel-Temp	MSE	0.70	0.60	0.56	0.53	0.57	0.60	0.51
	std. dev.	0.08	0.06	0.06	0.05	0.09	0.11	0.05

with low prediction accuracy. In our method we tried to obtain the best possible balance between the two.

3.4 Logical Rules

In this section we discuss the logical rules extracted from the steel-temperature dataset for three methods: the regression tree, the method of Setiono and our method. Because it would require about several pages to print out all the obtained rules, only some chosen examples are presented.

In the Setiono's method, first we have to calculate the position of signals on transfer functions generated by the network as a response to a given vector and obtain the coefficients by which each input feature will be multiplied and then we obtain a rule for each vector in the form:

y=0.297*x1+0.114*x2+0.018*x3+0.275*x4+0.077*x5-0.190*x6+...+0.197

The main problem is the coefficients are different depending on a given vector and they can take a lot of different values, even more than 100 for our dataset. That makes the rules too difficult to understand, especially in limited amount of time, as it is required in the production environment.

The logical rules generated by our method are similar to the rules generated by a regression tree. They are generated in a different way, but they have very similar form; first we decide to what cluster a particular vector belongs (the number of cluster is below 10) and then the final rule has the following form:

if (x1<0.207 and 0.334<x4<0.565) y=0.478*x3+0.232

The difference is that first the number of rule sets is below 10 and second that the rule has the crisp part, which is easier to understand and a very short linear part, which in first approximation maybe also replaced with a constant value.

4 Conclusions

The noticeable difference between a neural network and a decision tree is that the neural network considers all the attributes at each training stage. An univariate decision tree makes a choice of the most important attribute at every node according to some predefined criteria. As a result the split points are not always chosen inn a globally optimal way but in a way that is optimal for the current node and it is not guarantied that the sum of local maxima gives a global maximum. The next issue with decision trees are the predefined shapes of boundaries and the final mapping of leaves, even if done by linear regression models do not cover smoothly all the output space.

For that reason we decided to build the model based on a neural network and to overcome the neural network limitation of providing simple logical rules, especially for regression tasks, we incorporated the split idea known from decision trees. The rules generated by the model achieve only a little poorer accuracy than the best regression model, which we were able to build (committee of MLP networks), while the rule comprehensibility is comparable to that of decision trees with a significantly higher accuracy

at the same time. The system is currently being tested at the real technological data at one of polish steelworks for the purpose of steel temperature prediction in the electric arc furnace.

Acknowledgment. The work was sponsored by the Polish Ministry of Science and Higher Education, projects No. 4866/B/T02/2010/38 and 4421/B/T02/2010/38.

References

1. Corchado, E., et al.: Hybrid intelligent algorithms and applications. Information Science 180(14), 2633–2634 (2010)
2. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)
3. http://www.kordos.com/his.html
4. Blake, C., Keogh, E., Merz, C.: UCI Repository of Machine Learning Databases (1998-2011), http://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength
5. Blake, C., Keogh, E., Merz, C.: UCI Repository of Machine Learning Databases (1998-2011), http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime
6. Kordos, M., Blachnik, M., Perzyk, M., Kozłowski, J., Bystrzycki, O., Gródek, M., Byrdziak, A., Motyka, Z.: A Hybrid System with Regression Trees in Steel-Making Process. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS (LNAI), vol. 6678, pp. 222–230. Springer, Heidelberg (2011)
7. Kordos, M., Blachnik, M., Wieczorek, T., Golak, S.: Neural Network Committees Optimized with Evolutionary Methods for Steel Temperature Control. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part I. LNCS (LNAI), vol. 6922, pp. 42–51. Springer, Heidelberg (2011)
8. Kordos, M., Blachnik, M., Wieczorek, T.: Evolutionary Optimization of Regression Model Ensembles in Steel-Making Process. In: Yin, H., Wang, W., Rayward-Smith, V. (eds.) IDEAL 2011. LNCS, vol. 6936, pp. 369–376. Springer, Heidelberg (2011)
9. Setiono, R., Thong, J.: An approach to generate rules from neural networks for regression problems. European Journal of Operational Research 155(1) (2004)
10. Wang, J., et al.: Regression rules extraction from artificial neural network based on least squares. In: 7th Int. Conference on Natural Computation (ICNC), Shanghai (2011)
11. Saitoa, K., Nakano, R.: Extracting regression rules from neural networks. Neural Networks 15, 1279–1288 (2002)
12. Markowska-Kaczmar, U., Mularczyk, K.: GA-Based Rule Extraction from Neural Networks for Approximation. In: Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 141–148 (2006)
13. Kordos, M.: Neural Network Regression for LHF Process Optimization. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008. LNCS, vol. 5506, pp. 453–460. Springer, Heidelberg (2009)
14. Blachnik, M., Mączka, K., Wieczorek, T.: A Model for Temperature Prediction of Melted Steel in the Electric Arc Furnace(EAF). In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS, vol. 6114, pp. 371–378. Springer, Heidelberg (2010)
15. Duch, W., Setiono, R., Zurada, J.: Computational intelligence methods for understanding of data. Proceedings of the IEEE 92(5), 771–805 (2008)
16. Kordos, M., Duch, W.: Variable Step Search Algorithm for Feedforward Networks. Neurocomputing 71(13-15), 2470–2480 (2008)

Transfer Learning Approach to Debt Portfolio Appraisal

Tomasz Kajdanowicz, Slawomir Plamowski, Przemyslaw Kazienko, and Wojciech Indyk

Wroclaw University of Technology, Wroclaw, Poland
Faculty of Computer Science and Management
{tomasz.kajdanowicz,slawomir.plamowski, kazienko,wojciech.indyk}@pwr.wroc.pl

Abstract. Machine learning and data mining algorithms usually assume that the training and future data have the same distribution and come from the same feature space. However, in majority of real-world problems, this is not true. In case of Debt portfolio appraisal we have sufficient training data only in another domain of interest, namely in other portfolios. Therefore, only knowledge transfer from these portfolios in inference for new one is possible. In the paper we propose transfer learning and learning based on similarity methods, basing on similarity between training and testing datasets. The proposed approach is examined in real domain debt portfolio valuation.

Keywords: Transfer learning, dataset selection, distance measures, debt valuation, prediction, supervised learning.

1 Introduction

Supervised learning task is mainly focused on providing inference abilities derived from training data. Training data consist of a set of training examples, each composed of a pair of input features X and a desired output value Y. Therefore the main task is to analyse training data and produce an inferred function Φ that maps input to output, $\Phi : X \rightarrow Y$. If the output is discrete, the function Φ is called a classifier. Otherwise, in case of continuous output, it is called a regression. If function Φ maps to interrelated set of more than one values it is structured prediction or structured output learning algorithm. On the whole, the inferred function Φ should be able to predict the correct output value for any valid input object. This requires learning algorithm to be able to generalize basing on the training data.

However, as it may be expected, the availability of training datasets utilized in learning algorithms has a great influence on the generalization abilities. Sometimes in many real world applications, it is very expensive or even impossible to collect the needed training data to build the models. The traditional inference, based on previous learning in the same domain, is insufficient and in such case it is expected to use more sophisticated knowledge transfer or transfer learning between distinct tasks from similar domains.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 46–55, 2012.

The straightforward learning situation arises when learning concerns data from particular domain and describes always the same stationary object. The statistical dependencies between examples remain unchanged and training may be performed using the same source of training and testing data. Such data, as long as being of appropriate size, may deliver satisfactory generalisation abilities and no transfer learning is required. But in order to generalise from data describing non-stationary objects, learning algorithms are expected to model concept drift [7] phenomenon identified by changes in data probability distributions. As concept drift may be caused by changes of prior, conditional or posterior probabilities of data, appropriate methods must address the problem, among them these based on appropriate training set selection.

Another situation occurs when generalisation needs to be performed for objects for which training data is not available. In such case, learning is performed using data from the same domain but describing other similar objects. An example of such a situation are across-network classification where learning performed on one network adjust models used in generalisation on another network [8] or debt portfolio value prediction where value of appraisal of particular portfolio is done using other similar portfolios [5].

The paper considers the problem of transfer learning in the prediction task when inference is based on models learnt on data from this same domain but describing other similar objects. The paper presents a comparison between two learning techniques for that task: learning based on similarity and transfer learning.

Obviously, the greater similarity/smaller distance between objects used in learning and those the inference is applied to, the better performance of inference methods. Similarity/distance identification between training and testing objects can be reduced to similarity/distance measurement between datasets describing their input features, namely similarity/distance between X_{train} and X_{test}. Aforementioned similarity and distance can be invoked interchangeably as similarity can be measured by distance, i.e. two objects are similar if the distance is close to zero. In general, distance is defined as a quantitative degree of how far apart two objects are [2]. The choice of distance measure depends on the representation of objects and type of measurement. Training sets in supervised learning are usually represented by matrices in which columns denote attributes and rows - object instances. A single cell of such matrix contains a value of particular attribute for a given instance. Hence, the problem of learning based on similarity denotes a learning on selected training datasets based on measuring the distance between them and is actually a matrix distance based selection.

On the other hand, transfer learning provides additional ability to apply knowledge derived from external to current datasets for generalisation. The main concern denotes then discovering which knowledge can be transferred and how the knowledge from distinct models should be transferred across domains.

The rest of the paper is organised as follows. In section 2 various approaches of transfer learning and learning based on distance measures are enumerated. In order to provide a better perspective on the considered application problem,

section 3 presents a real-world transfer learning problem in debt portfolio value prediction. In section 4 two approaches to transfer learning and learning based on similarity are described. Evaluation of the impact on prediction accuracy using proposed methods is presented in section 5. Finally, section 6 summarises this work.

2 Related Work

In general, the bunch of methods, called here learning based on similarity, assumes that learning a generalisation model is done on training datasets of the same domain. These datasets are selected among all available domains. Training set selection from a set of available historical datasets based on the distance between particular testing set and training set may be considered using two equivalent approaches: as selection based on distance between matrices of non-equal size or, better, as calculating measure of goodness of fit between probability density functions. While calculation of probability density for discrete random variables is performed with respect to the counting measure over the sample space, the density of continuous random variables is given by the integral of this variable's density. This may imply problems as the exact density is not known and the empirical one can be obtained only. Literature proposes either the estimation of probability density function [11] or, simply, consideration of discrete and finite histogram of random variable [2,14]. The histogram can be considered then as a vector, i.e. coordinates in some space, and numerous distances proposed in the literature can be applied to compare two densities.

There exist a substantial number of distance measures derived from various fields such as computer science, information theory, mathematics, physics, or statistics, etc. Some of them that may be used in distance calculation are standard Euclidean distance and KullbackLeibler distance. For a v and w, a vector version of probability density functions of V and W matrices, with length of both vectors equal to d, Euclidean and KullbackLeibler distances are define as in equation 1 and 2, respectively.

$$dist(v, w) = \sqrt{\sum_{i=1}^{d} |v_i - w_i|^2} \tag{1}$$

In general, Euclidean distance measures shortest distance between two points as a length of line and belongs to L_p Minkowski family of distance measures. Applying Shannons concept of probabilistic uncertainty (entropy) Kullback-Leibler distance introduces the relative entropy, called information deviation [4], see equation 2.

$$dist(v, w) = \sum_{i=1}^{d} v_i \ln \frac{v_i}{w_i} \tag{2}$$

Obviously, distance measures presented above are only example ones and a proper choice of representative distance measure depends on the type of measured data and the measurement itself. For further list of distance measures please refer to [2,15].

The approach of calculating distances between vector version of probability density functions tends to be reasonable but requires estimation of probability density function and sometimes it might be troublesome.

The distance of two datasets might be computed by application of other concept - distance matrices. However, this is limited to situation when both datasets have the same size (number of rows) and, what is more, the mapping bijection that states the clear relation of corresponding data examples is known. As the size of compared distinct datasets may differ and the mapping between data examples is not known, it is not always possible to compute distance matrices.

Nevertheless, the distance between datasets may be calculated using matrix norms [9]. The matrix norms are defined in terms of well known vector norms and therefore, it can be said, they are induced by vector norms [16]. As some basic norms like (for a given matrix A) matrix 1-norm - returns maximum of A column sums, matrix ∞-norm - returns maximum of A row sums or matrix 2-norm - returns square root of largest eigenvalue of $A \times A$, more sophisticated once needs to be applied to characterize the matrix [9]. One of them can be Frobenius norm. This norm is the sum of the squares of the Euclidean norms of the matrix columns [9]. Thus it is able to model variability of the data. Investigating the literature we can see that norms are not the perfect solution to model distances between matrices.

On the other hand, transfer learning provides additional ability to learning system making it possible to recognize and utilize knowledge learned in previous tasks (datasets) to new tasks [10]. By that it is meant that transfer learning aims to extract the knowledge from several source tasks and apply the knowledge to a target task. In contrast to previously mentioned learning based on similarity, rather than learning models individually on selected datasets (tasks), transfer learning applies generalised knowledge of all known tasks obtained in single run learning. Figure 1 shows the difference between the learning in traditional and transfer learning approaches. As we can see, the first technique try to learn each task from scratch, while transfer learning utilizes knowledge from some previous tasks to perform an inference.

3 Debt Portfolio Value Prediction

Determining the value of debt portfolios and choosing those with the greatest revenue potential is of a great importance for debt traders. Economically crucial decisions are based on the amount of possible repayment of liabilities. As traders (both buyers and sellers) apply distinct collection processes, amount of receivables obtained may be different. This constitutes the area for trading and to establish a transaction price. Therefore debt portfolio assessment is a complex task. However, as far as machine learning is concerned, this problem may be understood as a prediction task that assesses the possible repayment value from all

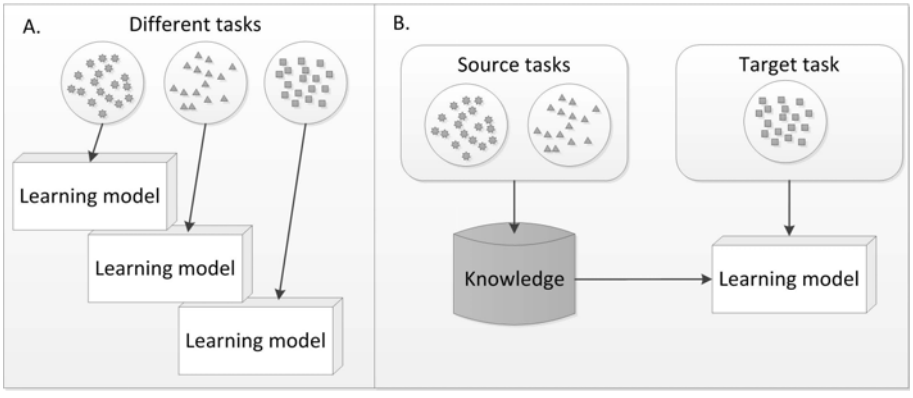


Fig. 1. Different Learning approaches: A – traditional machine learning, B – transfer learning

debt cases belonging to particular portfolio. The repayment is calculated based on historical data of debts.

The most common routine of debt portfolio trade starts when a seller, usually a bank, telecommunication company, etc. offers a set of debts, called debt package or portfolio, expecting a purchase proposal from buyers. Purchasers, usually a specialized debt recovery entities, offer price and the most suitable offer is chosen. The price proposed by a particular buyer may be obtained in variety of ways, among which the utilization of historical data of debt recovery in order to build a prediction model seems the most reasonable one. Such model provides an estimation of possible return from the package.

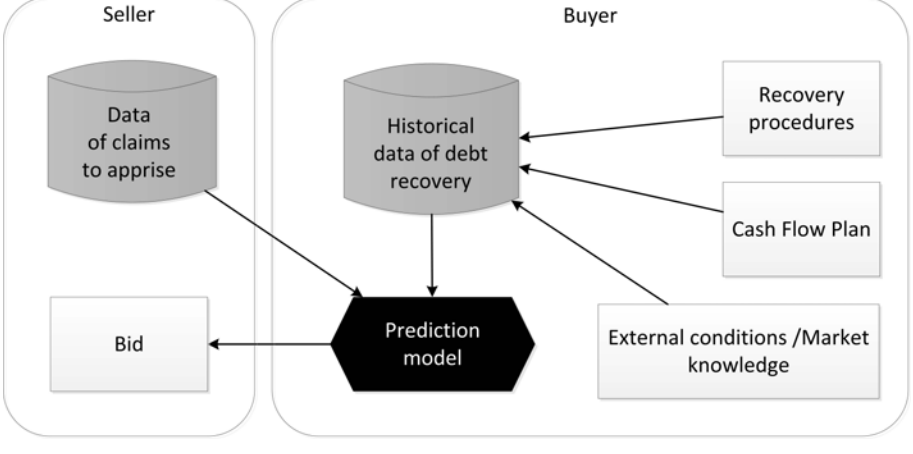


Fig. 2. The process of debt portfolio purchase with utilization of prediction model

In considered situation the valuation of debt portfolio is based on the data of historical claims with their repayment profiles over time. A debt collection company usually assumes that gathered repayment data reflects all important dependencies influencing repayment results like recovery procedures, cash flow plans and other external conditions. Such assumption simplifies the problem as changes in the probabilities caused by evolving business environment are ignored. The model trained on historical data is applied to predict the repayment amount of the offered portfolio. Based on the obtained results, bids are offered to the seller. The process of debt portfolio valuation for bid proposal is presented in Figure 2.

Summarizing, the most significant and sensitive part of debt trade is repayment value prediction process. The accuracy of prediction for offered portfolio relies mainly on model generalization capabilities and quality of training data. As it is very difficult to provide prediction using whole, large historical data, some training dataset selection mechanism needs to be employed. In the further part of the paper we present the method for training set selection for model learning, that is applied to considered business scenario.

4 Learning Based on Similarity and Transfer Learning Techniques for Debt Portfolio Appraisal

Assuming that training set can be treated as a matrix, the problem of training set selection is equivalent to the matrix selection using some notion of distance. Assuming additionally that the environment remains stationary, the generalisation can be done on the basis of historical datasets of the same domain, which describe similar objects with the same attributes. In the aforementioned debt prediction problem, historical dataset consist of debt portfolios, that have already been repaid. They are used to predict the repayment value of unknown, new portfolio. The learning could be done using all historical datasets, but from the practical point of view it would not always be possible (e.g. massive training data) and of high quality (poor inference from complex and non-discriminative data). Therefore some hybrid learning methods need to be applied.

Hereby we propose a method based on transfer learning concept, which one can describe as using knowledge from previous prediction tasks to acquire new knowledge in current task. One can make an assumption that aforetime acquired knowledge can be not only helpful, but also essential for future prediction. Such consideration can provide additional latent information that are transferred during training process.

The proposed method is based on the assumption that there exists a set T of k train sets A_i, $i \in \{1, \ldots, k\}$, $k \in \mathbb{N}$ and a single test set B. The actual task is to create a ranking of distances between each set A_i from T and test set B. Having created such ranking, train sets A_i are sorted in ascending order of distances. In the next step, the method of learning based on similarity utilizes closest A_i

sets for training procedure, whereas transfer learning generalizes all datasets and from now one is able to transfer the knowledge to new tasks, see differences in algorithms 1 and 2. It means that inference does not require training the models. Each dataset is generalised by one learning model and in order to balance consume the knowledge from multiple tasks, weights vector is calculated as sum of distances between considered train sets A_i and test set B, namely $dist(A_i, B)$, divided by the sum of distances, see equation 3. Trained predictors are then used to obtain results from testing set B (algorithm 2).

$$weight\_vector_i = \frac{dist(A_i, B)}{\sum_{k=1}^{M} dist(A_k, B)}, \tag{3}$$

In proposed method, separate prediction algorithm is used for each train set A_i. Afterwards we use these predictors to infer targets on test set B. Results of inference from different tasks are weighted by the aforementioned weights vector.

Algorithm 1. The pseudo code of learning and inference phase of the method for learning based on similarity

Require: set T of k training sets A_i, $i \in \{1, \ldots, k\}$, testing set B
1: **for all** training sets $A_i \in T$ **do**
2: calculate $dist(B, A_i)$
3: **end for**
4: build a distance ranking
5: select training dataset(s) using ranking
6: build model on selected dataset(s)
7: **return** inferred targets for B

Algorithm 2. The pseudo code of learning phase of the transfer learning approach

Require: set T of k training sets A_i, $i \in \{1, \ldots, k\}$
1: **for all** training sets $A_i \in T$ **do**
2: learn the model
3: **end for**
4: **return** set of models

5 Experiments and Results

The main objective of performed experiments was to test and evaluate the proposed transfer learning technique in debt appraisal task. Among others some standard performance measures were observed: Relative Error(RE), Mean Square Error (MSE), Coefficient of Correlation (R), Variance Accounted For (VAF), Maximum Absolute Error (MAE), Coefficient of Efficiency (COE).

Algorithm 3. The pseudo code of inference phase of the transfer learning approach

Require: set T of k training sets A_i, $i \in \{1, \ldots, k\}$, testing set B, set of learnt models Φ

1: **for all** training sets $A_i \in T$ **do**
2: calculate $dist(B, A_i)$ and $weight\_vector_i$
3: **end for**
4: infer targets for B using Φ and $weight\_vector_i$
5: **return** inferred targets for B

Experiments were carried out on fifteen 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 procedure of experiment accomplishes a prediction of possible repayment values for a B debt portfolio. Depending on learning approach, from among all or selected known portfolios learning sets are constructed. Using selected packages, the regression algorithms are trained and eventually basic tests for portfolio B are performed.

Based on described procedure, three distinct experimental scenarios are created. They vary in the number of selected portfolios for training and in utilized inference method, namely learning based on training set similarity and transfer learning. Therefore the best known methods from authors' previous findings [6] are compared with the transfer learning approach in the experiments. The first scenario uses the closest package for learning, the second – three closest packages and the third all packages but with distinct inference procedure. From this point, these scenarios are denoted as: *C, C3, TL* respectively. For examined scenarios Friedman test is performed. Results are presented in Table 1.

Table 1. Average rank positions determined in Friedman test

Measure/Rank	1^{st}	2^{nd}	3^{rd}
RE	TL (1.67)	C3 (2.13)	C (2.20)
MSE	C3 (1.47)	C (1.87)	TL (2.67)
R	C (1.53)	TL (2.00)	C3 (2.47)
VAF	TL (1.47)	C (1.93)	C3 (2.6)
MAR	TL (1.67)	C3 (1.93)	C (2.4)
COE	TL (1.33)	C (2.13)	C3 (2.53)

For each prediction algorithm statistical ranking is created to indicate optimal approach. We incorporated Friedman statistical test as intuitive and convenient procedure for different used approaches comparison. Mean rank position for each combination of method and scenario is shown in parentheses. The lower rank value, the lower observed performance measure yielded by prediction process.

As shown in Friedman test, usage of transfer learning approach results in better performance than using single and multiple closest datasets for training.

The results in Table 1 can be read as follows: for fifteen debt evaluation tasks selecting transfer learning approach, denoted by TL, results in the smallest mean squared Relative Error (RE), Variance Accounted For (VAF), Maximum Absolute Error (MAE) and Coefficient of Efficiency (COE). Friedman's test places this learning approach in the first place of ranking.

As it can be observed in Table 1 the TL approach performs worse in Mean Square Error (MSE) and Coefficient of Correlation (R) measures in comparison with other methods. However, according to the nature of debt portfolio evaluations the objective is to minimize the Relative Error (RE). Therefore some approaches may be better in MSE minimization while it is not a main target.

6 Conclusions

The problem of transfer learning was considered in the paper. We introduced a learning based on similarity method, that selects training sets to be used in training. Sets are chosen based on distance between two datasets. Moreover, we proposed transfer learning approach to this same task. Transfer learning method does not require model learning each time the inference needs to be employed.

The proposed methods were examined on real datasets in the debt portfolio valuation domain. The results indicated that proposed transfer learning method can be used to infer effectively in the debt portfolio appraisal domain.

Further experimentation will consider a comparison of the presented method with other approaches. Moreover, further studies will focus on discovery and description of properties of proposed method.

Acknowledgement. This work was partially supported by The Polish Ministry of Science and Higher Education the research project 2011-2012, 2011-2014 and Fellowship co-financed by The European Union within The European Social Fund.

References

1. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study. IEEE Transactions on Evolutionary Computation 7(6), 561–575 (2003)
2. Cha, S.H.: Comprehensive survey on distance/similarity measures between probability density functions. International Journal of Mathematical Models and Methods in Applied Sciences 1(4), 300–307 (2007)
3. Coifman, R.R., Wickerhauser, M.V.: Entropy-based algorithms for best basis selection. IEEE Transactions on Information Theory 38, 713–718 (1992)
4. Deza, E., Deza, M.M.: Dictionary of Distances. Elsevier (2006)
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., Plamowski, S., Kaznieko, P.: Training Set Selection Using Entropy Based Distance. In: The IEEE Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2011, pp. 340–344. IEEE Computer Society (2011)

7. Kurlej, B., Wozniak, M.: Active learning approach to concept drift problem. Logic Journal of the IGPL (2011), doi:doi:10.1093/jigpal/jzr011
8. Lu, Q., Getoor, L.: Link-based classification using labeled and unlabeled data. In: ICML 2003 Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining (2003)
9. Meyer, C.D.: Matrix analysis and applied linear algebra. Society for Industrial and Applied Mathematics (2000)
10. Pan, S.J., Yang, Q.: A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (2010)
11. Rencher, A.: Methods of multivariate analysis. John Wiley & Sons (2002)
12. Son, S.-H., Kim, J.-Y.: Data Reduction for Instance-Based Learning using Entropy-Based Partitioning. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3982, pp. 590–599. Springer, Heidelberg (2006)
13. Theodoris, S., Koutroumbas, K.: Pattern Recognition. Elsevier (2009)
14. Toussaint, G.T.: Bibliography on estimation of misclassification. IEEE Transactions on Information Theory 20(4), 472–479 (1974)
15. Ullah, A.: Entropy, divergence and distance measures with econometric applications, Department of Economics. University of California - Riverside (1993)
16. Zhou, K., Doyle, K., Glover, K.: Robust and Optimal Control. Prentice-Hall (1996)

Generalized Weighted Majority Voting with an Application to Algorithms Having Spatial Output

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 using a generalization of the weighted majority voting scheme to locate the optic disc (OD) in retinal images automatically. The location with the maximal sum of the weights of OD center candidates falling into a disc of radius predefined in the clinical protocol is chosen for optic disc. We have worked out a weighted voting scheme, where besides the weights, an additional (e.g. geometrical) condition has to be taken into account in making the final decision. We can achieve better overall performance with this generalized weighted voting system than with the weighted majority voting and each individual algorithm.

Keywords: Biomedical imaging, diabetic retinopathy, classifier combination, majority voting, weighted voting.

1 Introduction

Diabetic retinopathy (DR) is an eye disease (damage to the retina) that is the most frequent cause of new cases of blindness. In automatic grading of the retinal images and in making diagnosis, determining the exact location of the main anatomical features (e.g. the optic disc and the macula) is among the first steps. Both the optic disc and the macula can be considered as a circular region (disc) on the retinal images. The optic disc is very bright, while the macula is a highly pigmented spot whose center is called fovea which is responsible for the sharpest vision.

In our approach, we organize several different individual OD detector algorithms into a weighted voting system to raise the accuracy of the OD detection ([1],[2]). Each OD algorithm results in a single pixel as output for the OD center. In our application for the OD detection, the majority voting cannot be applied directly since the spatial replacement of each vote also counts in making the final decision. In the generalized weighted voting system, the OD center candidate of

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 56–67, 2012.

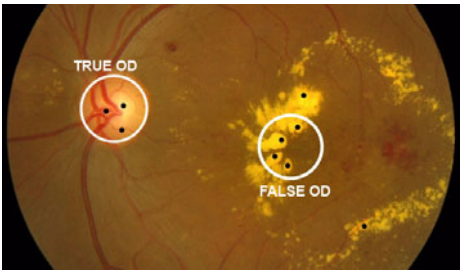


Fig. 1. Results of the different OD detecting algorithms

each detector has been combined and the minimal bounding circles for all subgroups of the candidates are considered. The radius of the circle must be less than or equal to the radius of the optic disc that is a clinically predetermined constant. In this weighted voting system we choose the circle with the maximal sum of the weights assigned to the candidates falling inside this circle.

Weighted majority voting is widely examined in the literature (see e.g. [3], [4]). For characterizing the accuracy of the weighted system to our application a corresponding theoretical model is needed. If we consider the bounding circle with the maximal weight sum, then similarly to a traditional weighted majority setup, we can make a good decision even in the case when the bad candidates have pure majority in number. In our former work [5], we have generalized the classical majority voting to our problem. Now, just as in the traditional case, we check how weighted majority can outperforms classical majority voting. In the non-weighted generalized voting system 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 such as in the case shown in Fig. 1. In the weighted generalized voting system we make a wrong decision only in that case when a subset of bad candidates having larger sum of weights than the sum of weights assigned to the good ones can be bounded by a circle with an appropriate radius. In the case demonstrated in Fig. 1. good decision is made applying the weighted generalized voting system.

These observations motivated us to work out a corresponding theoretical model, where bad votes can overcome good ones only if a further (e.g. geometrical) condition is fulfilled. This additional condition is the spatial closeness of the candidates in the above application. With this model we generalize the classical non-weighted and weighted majority voting scheme, since in the case of less good votes we may 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 recall our results for the generalization of the non-weighted voting system, while Section 4 discusses the weighted majority voting and our

generalized weighted system. In Section 5, our experimental results for the specific OD detection application are presented. Section 6 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 $\Omega = (\omega_1, \omega_2, \ldots, \omega_c)$ is a set of class labels. If the classifier decisions are combined in the majority voting, then the class label ω_i is assigned to $\mathbf{x}$ that is supported by the majority of the classifiers D_i. In the case of a tie, the decision is usually made randomly.

As a special case, we can consider binary classifiers examined exhaustively in the literature. Let $n (n \in \mathbb{N})$ be odd, $\Omega = (\omega_1, \omega_2)$ (that is, each classifier output is a binary vector) and all classifiers have the same classification accuracy $p (p \in [0, 1])$. An accurate class label is given by the majority vote if at least $[n/2]$ classifiers give correct answers. The overall accuracy of correct classification in majority voting with independent classifier decisions can be computed by the binomial formula:

$$P = \sum_{k=0}^{[n/2]} \binom{n}{k} p^{n-k} (1-p)^k. \tag{1}$$

If the classifiers are independent and $p > 0.5$, then this method is guaranteed to outperform the individual classifiers. Applying the majority voting in pattern recognition, several interesting results can be found in [6] (e.g. about adding one or two new classifiers to the voting system).

3 The Generalized Majority Voting

The classifiers making independent errors are generally considered independent, so under this assumption, the error of the classifiers can be modelled by random variables and their distributions. If we assume initially equal probabilities of errors for all classifiers, the model with Bernoulli distribution is the simplest and for this case the most appropriate one.

In this section, we recall and slightly re-formulate the theoretical and experimental results for the generalized majority voting system [5]. Let $\eta = (\eta_1, \ldots, \eta_n)$ be an n-dimensional random variable (n classifiers). 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 $p \in [0, 1]$ (each classifier has the same accuracy p). Execute the experiment η independently t times (t objects to be classified), and write the outcomes (outputs of the classifiers) 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 1 values (k correct classification for the j-th object) then let

$$P(\chi_j = 1) = p_{nk}, \quad P(\chi_j = 0) = 1 - p_{nk} \quad (j = 1, \ldots, t) \ , \tag{3}$$

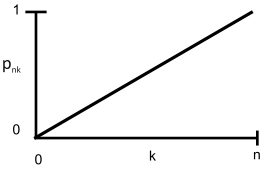
where the p_{nk}-s ($k = 0, 1, \ldots, n$) are given numbers with monotone increasing property fulfilled in all rows: $0 \leq p_{i0} \leq \cdots \leq p_{ii} \leq 1 \quad (i = 1, \ldots, n)$. The p_{nk} describes the probability of the good final decision in case of k correct classifications from n classifiers.

Observe that the χ_j-s are independent. Finally, put $\xi = |\{j \ : \ \chi_j = 1\}|$, that is, ξ is the number of the good final decisions for t objects. We observe that all the individual decisions η_i ($i = 1, \ldots, n$) are of binomial distribution with parameters (t, p). Then we get that ξ has also binomial distribution, with the appropriate parameters (t, q), where

$$q = \sum_{k=0}^{n} p_{nk} \binom{n}{k} p^k (1-p)^{n-k}. \tag{4}$$

In order to the generalized majority voting outperform the individual decisions, we need only to guarantee that $q \geq p$.

In that case when p_{nk} is linear in k for a given n, that is $p_{nk} = k/n$ ($k = 0, 1, \ldots, n$), then we get $q = p$.



(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

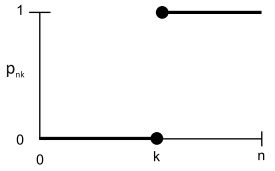
Fig. 2. The results of the linear case

If we suppose that $p_{nk} \geq k/n$ for all $k = 0, 1, \ldots, n$, then $q \geq p$, so in this case the generalized majority voting outperforms the individual decisions.

As a special case of the generalized majority voting, when 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, we get the classical majority voting.

We indicate the overall performance P of the voting system in Fig. 2. and Fig. 3. for different L number of classifiers/algorithms at different classifier accuracies p in the linear case when $p_{nk} = k/n$ and in the classical majority voting case, respectively.

We give another example of the matrix p_{nk} that is motivated by our application for OD detection. In this case, the behavior of p_{nk} as a function of k for a fixed n and the system accuracy are illustrated in Fig. 4.

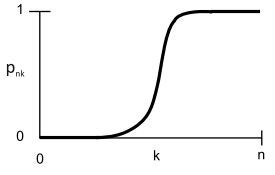


(a) The curve of p_{nk}

	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. 3. The results of the classical majority voting scheme



(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. 4. The results of our application for OD detection

From Fig. 4, we can see, that for a fixed n, p_{nk} increases exponentially in k. This follows from the results of [7] about the diameter d of a point set. The probability that d is not less than a given constant decreases 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.

4 Modifications on the Decision Rule

In this section, we modify the final decision rule of the ensemble which will result in further improvement of the system accuracy. Our generalization is based on the assignment of weights to the ensemble members (classifiers). First, we recall the necessary procedure for finding the weights in classical majority voting (see e.g. [4]). Then, we derive how the optimal weights can be found for our generalized voting case.

4.1 Weighted Voting System

For weighted voting system, first let us consider the classifiers $(D_1, D_2, \ldots, D_n)$ with accuracies $(p_1, p_2, \ldots, p_n)$, respectively. Then, let $d_{i,j}$ be defined in the following way: $d_{i,j} = 1$, if the classifier D_i labels $\mathbf{x}$ in the class ω_j, and $d_{i,j} = 0$, otherwise. In case of weighted voting, the discriminant function for class ω_j is given as:

$$g_j(\mathbf{x}) = \sum_{i=1}^{n} b_i d_{i,j}, \tag{5}$$

where the weight b_i corresponds to the classifier D_i. Note that the following discriminant functions are equivalent for the given decision rule:

$$g_j(\mathbf{x}) = P(s|\omega_j)P(\omega_j), \quad g_j(\mathbf{x}) = \log(P(s|\omega_j)P(\omega_j)), \tag{6}$$

where $s = [s_1, \ldots, s_n]$ is the vector with the label output of the ensemble. Here $s_i \in \Omega$ is the label suggested for $\mathbf{x}$ by the classifier D_i and $P(\omega_j)$ is the prior probability for class ω_j.

In a weighted majority voting system, the class label ω_k is chosen for $\mathbf{x}$ if

$$g_k(\mathbf{x}) = \max_{j=1,\ldots,n} g_j(\mathbf{x}) = \sum_{i=1}^{n} b_i d_{i,k}. \tag{7}$$

In a weighted majority system a natural question is that how to choose the optimal weights for the classifiers. If we consider independent classifiers, then the system accuracy is maximized by assigning weights (see e.g. [4]):

$$b_i \propto \log \frac{p_i}{1-p_i}, i = 1, \ldots, n. \tag{8}$$

Note that, conditional independence is assumed here, that is:

$$P(s|\omega_j) = \prod_{i=1}^{n} P(s_i|\omega_j), \tag{9}$$

where $s = [s_1, \ldots, s_n]$ is the same as above.

The weights $b_i \propto log\frac{p_i}{1-p_i}$ do not guarantee the minimum classification error, because the prior probabilities for the classes $P(\omega_j)$ have to be taken into account, too. More precisely, if the individual classifiers are independent, and the a priori likelihood is that each choice is equally likely to be correct, then the decision rule that maximizes the system accuracy is a weighted majority voting rule obtained by assigning weights $b_i \propto \log \frac{p_i}{1-p_i}$.

In contrast to the classical majority voting, we equip each classifier output with different weights b_i, where $0 \leq b_i \leq 1$ $(i = 1, \ldots, n)$. It seems natural to give the classifiers with larger accuracies larger importance in making the final decision. Note that the classical majority voting scheme can be considered as a special case of the weighted voting system since in the majority rule the weight of each vote given by a classifier is constrained to be $b_i = 1$ for all $i = 1, \ldots, n$.

4.2 Generalized Weighted Voting System

We can also assign weights to the classifiers within our generalized voting scheme presented in Section 3. If we consider the classifiers $(D_1, D_2, \ldots, D_n)$ with respective accuracies $(p_1, p_2, \ldots, p_n)$ and weights $b_1, \ldots, b_n$, then the final decision is made by choosing the maximal sum of weights, where some additional constraints (e.g. a geometrical one for OD detection) have to be fulfilled by the classifier outputs. Let us consider the probability $(1 - p_i)r_i$ with $r_i \in [0, 1]$ for

the i-th classifier that means that the i-th classifier makes wrong classification and participates in making a bad decision.

In our application, we choose the maximal sum of those weights of the algorithms whose outputs can be bounded by a circle with an appropriate radius. An algorithm takes part in making a bad decision if its output falling outside the optic disc meets other bad candidates. For the algorithm D_i with accuracy p_i giving a bad candidate (x_i, y_i) for the optic disc, we consider that the distribution of (x_i, y_i) is uniform outside the optic disc for all i $(i = 1, \ldots, n)$. In this case, we have:

$$r_1 = \ldots = r_n = \frac{T_0}{T - T_0}, \tag{10}$$

where T_0 and T are the area of the optic disc and the ROI (whole useful image domain), respectively, so r_i is the same predetermined constant for all i $(i = 1, \ldots, n)$. For better understanding, see Fig. 5, where we show how bad candidates can fulfil the geometric constraint by falling inside a disc with OD radius.

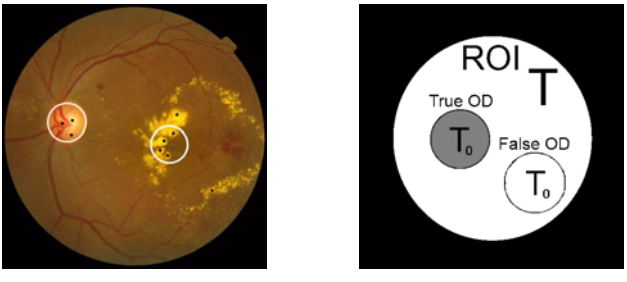


Fig. 5. Retinal image and a schematic one with the used notation

The next theorem gives the answer on how to select the weights in our generalized weighted majority voting model.

Theorem 1. *If independent classifiers $(D_1, D_2, \ldots, D_n)$ are given (conditional independence is assumed), then the optimal weight b_i for the classifier D_i with accuracy p_i can be calculated as:*

$$b_i \propto \log \frac{p_i}{(1 - p_i)^2 r_i (1 - r_i)}. \tag{11}$$

Proof. Let $s = [s_1, \ldots, s_n]$ denote the vector with the label output of the ensemble, where $s_i \in \Omega$ is the label suggested for $\mathbf{x}$ by the classifier D_i. A Bayes-optimal set of discriminant functions based on the outputs of the n classifiers is

$$g_j(\mathbf{x}) = \log P((\omega_j) P(s|\omega_j)), \quad (j = 1, \ldots, c). \tag{12}$$

From the conditional independence, for the discriminant functions $g_j(\mathbf{x})$ we get

$$\log P(\omega_j)P(s|\omega_j) = \log\left[P(\omega_j)\prod_{i=1}^{n} P(s_i|\omega_j)\right] = \tag{13}$$

$$\log P(\omega_j) + \log\left(\prod_{i,s_i=\omega_j} P(s_i|\omega_j)\prod_{i,s_i\neq\omega_j} P(s_i|\omega_j)\right) = \tag{14}$$

$$\log P(\omega_j) + \log\left(\prod_{i,s_i=\omega_j} p_i \prod_{i,s_i\neq\omega_j} (1-p_i)r_i \prod_{i,s_i\neq\omega_j} (1-p_i)(1-r_i)\right) = \tag{15}$$

$$\log P(\omega_j) + \log\left(\prod_{i,s_i=\omega_j} \frac{p_i(1-p_i)}{1-p_i} \prod_{i,s_i\neq\omega_j} (1-p_i)r_i \prod_{i,s_i\neq\omega_j} (1-p_i)(1-r_i)\right) = \tag{16}$$

$$\log P(\omega_j) + \log\left(\prod_{i,s_i=\omega_j} \frac{p_i}{1-p_i} \prod_{i,s_i\neq\omega_j} (1-p_i)r_i(1-r_i)\prod_{i=1}^{n}(1-p_i)\right) = \tag{17}$$

$$\log P(\omega_j) + \sum_{i,s_i=\omega_j} \log\frac{p_i}{1-p_i} + \sum_{i,s_i\neq\omega_j} \log((1-p_i)r_i(1-r_i)) + \sum_{i=1}^{n}\log(1-p_i). \tag{18}$$

The last term does not depend on the class label j so we can reduce the discriminant function to

$$g_j(\mathbf{x}) = \log P(\omega_j) + \sum_{i,s_i=\omega_j} \log\frac{p_i}{1-p_i} + \sum_{i,s_i\neq\omega_j} \log((1-p_i)r_i(1-r_i)) = \tag{19}$$

$$\log P(\omega_j) + \sum_{i=1}^{n} d_{i,j}\log\frac{p_i}{1-p_i} + \sum_{i=1}^{n}(1-d_{i,j})\log((1-p_i)r_i(1-r_i)) = \tag{20}$$

$$\log P(\omega_j) + \sum_{i=1}^{n} d_{i,j}\log\frac{p_i}{(1-p_i)^2 r_i(1-r_i)} + \sum_{i=1}^{n}\log((1-p_i)r_i(1-r_i)). \tag{21}$$

The last term of the summation is also independent from the class label j so it can be omitted. If we consider the equations:

$$g_j(\mathbf{x}) = \log P(\omega_j) + \sum_{i=1}^{n} d_{i,j}\log\frac{p_i}{(1-p_i)^2 r_i(1-r_i)}, \tag{22}$$

and

$$g_j(\mathbf{x}) = \sum_{i=1}^{n} b_i d_{i,j}, \tag{23}$$

we get that the weights:

$$b_i \propto \log\frac{p_i}{(1-p_i)^2 r_i(1-r_i)} \tag{24}$$

that are supposed to maximize the system accuracy.

Note that, similarly to classical majority voting, the weights given in (24) do not always guarantee the minimum classification error. Only if the individual classifiers are independent and the prior probabilities for the classes $P(\omega_j)$ are equal, the decision rule that maximizes the system accuracy is a weighted majority voting rule, obtained by assigning the above weights.

4.3 Weighted Majority Voting in OD Detection

In our application, the output of each OD detecting algorithm is the OD center given as a single pixel with coordinates (x_0, y_0). In our ensemble-based system we have the set of class labels $\{\omega_{(x,y)} | (x, y) \in ROI\}$. For an OD detector (as a classifier) with its output (x_0, y_0), the class label $\omega_{(x_0,y_0)}$ is assigned to the detector. In other words, the classifier voted to the pixel (x_0, y_0) as OD center. The classification is considered to be correct if the output (x_0, y_0) falls inside the true optic disc on the retinal image. We can define the decision rule as the sum of the weights of the OD detecting algorithms, whose outputs can be bounded by a circle of the OD radius. Such a circle with the maximal sum of weights is accepted as the final decision for the OD.

In this application, the condition for the equal prior probabilities for the classes is fulfilled if we suppose uniform distribution of the candidates both inside and outside the optic disc.

In contrast to the non-weighted systems, less conflicting situations can be obtained when the decision is not exact because of the equal number of outputs falling inside the discs of the predetermined radius. Further improvement of this weighted system on majority voting is that there is no need for accuracy constraints $p > 0.5$ on individual algorithms to achieve larger system accuracy. It can be shown that this weighted voting rule always outperforms the classical majority rule because in case of a conflict (when the same number of votes are densified in different discs of a given radius) majority rule decides randomly between the disc candidates, while the weighted voting system can handle the conflict determining to the sum of the weights corresponding the output votes falling inside the discs.

5 Experimental Results

We compare the system accuracies of the classical and the weighted majority voting for different accuracies and different weights. In our tests, we considered three different types of accuracies for the algorithms:

- $A_1 : p_1 = p_2 = \ldots = p_9 = 0.6$,
- $A_2 : p_i = 1 - 0.1i$, $i = 1, \ldots, 9$,
- $A_3 : p_1 = 0.6472, p_2 = 0.9765, p_3 = 0.3205, p_4 = 0.7593, p_5 = 0.3153$, $p_6 = 0.2276, p_7 = 0.9582, p_8 = 0.7671, p_9 = 0.6432$.

The case A_1 is often examined in the literature with equal weights, A_2 is a theoretical example, while A_3 contains true accuracies of OD detecting algorithms measured on the Messidor test database[1] containing 1200 retinal images.

For comparative studies, we apply the following weights b_i for the i-th algorithm having accuracies p_i $(i = 1, \ldots, 9)$:

- $B_1 : b_i = p_i$,
- $B_2 : b_i = log\frac{p_i}{1-p_i}$,
- $B_3 : b_i = \frac{p_i}{(1-p_i)^2 r_i(1-r_i)}$.

That is, in case B_1 each weight is equal to the accuracy of the individual algorithm (such as taken the i-th algorithm with accuracy p_i, then it participates in the final decision with weight $b_i = p_i$). B_2 is suggested as optimal for the classical weighted majority voting, while B_3 is the proposed assignment for our generalized weighted majority voting. In this way, we give a practical example to confirm the theoretical derivation of the optimal weights given in Section 4.2.

We apply OD detecting algorithms as classifiers, so we can test and compare the overall performance of the different voting systems on classifier output generated artificially. In lack of independent OD detecting algorithms providing these accuracies, we are not able to test and compare the voting systems on retinal images. We generate the classifier outputs in the following way: we consider a disc of radius R (ROI) and a disc of radius R_0 inside the ROI (optic disc), where $R = 712$ and $R_0 = 48$ pixels, respectively. We generate 9 output points with coordinates (x_i, y_i) (as outputs the D_i's), where the probability that the point (x_i, y_i) falls inside the optic disc is p_i and the distribution of (x_i, y_i) is uniform outside the optic disc. Now, the probability r_i $(i = 1, \ldots, 9)$ can be determined as:

$$r_1 = \ldots = r_n = \frac{T_0}{T - T_0} = \frac{R_0^2}{R^2 - R_0^2}. \tag{25}$$

In this test we compare the performance of the following voting systems: MV- majority voting, WMV- weighted majority voting, GMV- generalized majority voting, WGMV- weighted generalized majority voting. The system accuracies for the individual accuracy setups A_1, A_2, A_3 with the weight assignments (B_1, B_2, B_3) are given in Fig. 6(a), Fig. 6(b), Fig. 6(c), respectively.

From the tables we can see that if all weights are equal, then it naturally results in the same system accuracy as the non-weighted voting scheme, otherwise, weighted voting outperforms non-weighted voting. Our generalized non-weighted (weighted) voting system has better overall performance than the classical non-weighted (weighted) majority voting scheme.

For the OD detection application, we can test and compare our generalized non-weighted and generalized weighted voting system on a real database of retinal images, as well. The Messidor dataset [1] considered for this aim contains 1200 retinal images. In this test, we assigned the optimal weights derived in Section 4.2 to the participating algorithms (classifiers) having individual accuracies $p_1 = 0.6472, p_2 = 0.9765, p_3 = 0.3205, p_4 = 0.7593, p_5 = 0.3153,$

[1] http://messidor.crihan.fr

A_1	MV	WMV	GMV	WGMV
B_1	0.7323	0.7323	0.9948	0.9996
B_2	0.7380	0.7380	0.9941	0.9991
B_3	0.7326	0.7326	0.9948	0.9989

(a) System accuracies for the set A_1

A_2	MV	WMV	GMV	WGMV
B_1	0.5012	0.8066	0.9889	0.9943
B_2	0.4965	0.9688	0.9901	0.8712
B_3	0.5009	0.7289	0.9877	0.9951

(b) System accuracies for the set A_2

A_3	MV	WMV	GMV	WGMV
B_1	0.8241	0.9526	0.9996	1.0000
B_2	0.8260	0.9926	0.9989	0.9941
B_3	0.8258	0.9481	0.9989	0.9998

(c) System accuracies for the set A_3

Fig. 6. Overall system accuracies for the set of classifier accuracies

$p_6 = 0.2276, p_7 = 0.9582, p_8 = 0.7671, p_9 = 0.6432$ (as given in case $A3$). However, note that we have no information about the dependencies among these algorithms. Despite the unknown dependencies of the algorithms, we found that weighted majority voting with its system accuracy 0.98 outperformed classical majority voting (system accuracy 0.974), and also all the individual accuracies.

6 Conclusion and Future Plans

We have introduced a new theoretical model that enables the investigation of majority voting systems being more general than the classical majority voting scheme. As for practice, we apply this generalization to set up ensemble of algorithms providing spatial output. This generalized voting system (when some additional geometrical constraints have to be fulfilled) can be applied in that case when weights are assigned to the classifiers, as well. In our specific application, larger overall system accuracy is achieved, than in the case of individual algorithms and weighted voting outperformed the non-weighted one. Same results can be expected for similar image processing problems, where the algorithms vote with a single pixel or region. In our application, adding a new independent algorithm to the system seems to be very effective because of the exponential behavior of the system accuracy. The full characterization of the participating algorithms to achieve the best system performance is still an open issue.

A further issue regarding 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 our system, it will be a future research direction to see how the accuracy can be raised by removing/adding algorithms from/to the existing system in consideration to individual accuracies and dependencies.

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. 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)
2. Qureshi, R.J., Kovacs, L., Harangi, B., Nagy, B., Peto, T., Hajdu, A.: Combining Algorithms for Automatic Detection of Optic Disc and Macula in Fundus Images. Computer Vision and Image Understanding 116(1), 138–145 (2012)
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. Toman, H., Kovacs, L., Jonas, A., Hajdu, L., Hajdu, A.: A Generalization of Majority Voting Scheme for Medical Image Detectors. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS (LNAI), vol. 6679, pp. 189–196. Springer, Heidelberg (2011)
6. 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)
7. Appel, M.J., Najim, C.A., Russo, R.P.: Limit Laws for the Diameter of a Random Point Set. Adv. in Appl. Probab. 34(1), 1–10 (2002)
8. Altincay, H.: On Naive Bayesian Fusion of Dependent Classifiers. Pattern Recognition Letters 26, 2463–2473 (2005)

Towards the Reduction of Data Used for the Classification of Network Flows

Maciej Grzenda

Warsaw University of Technology,
Faculty of Mathematics and Information Science,
00-661 Warszawa, Pl. Politechniki 1, Poland
M.Grzenda@mini.pw.edu.pl
Orange Labs Poland
02-691 Warszawa, ul. Obrzeżna 7, Poland
Maciej.Grzenda@telekomunikacja.pl

Abstract. The ever growing volume of network traffic results in the need for even more efficient data processing in Intrusion Detection Systems. In particular, the raw network data has to be transformed and largely reduced to be processed by data mining models.

The primary objective of this work is to control the dimensionality reduction (DR) of network flow records in view of the accuracy of misuse detection. A real data set, containing flow records with potential spam messages, is used to perform the tests of the proposed method. The algorithm proposed in this study is applied to investigate the merits of hybrid models composed of dimensionality reduction, neural networks, and decision trees. The benefits of dimensionality reduction and the impact of the process on the overall spam detection rates and false positive rates are investigated. The advantages of the proposed technique over standard a priori selection of reduced dimension are discussed.

Keywords: Network flows, spam detection, dimensionality reduction, multilayer perceptron, decision tree.

1 Introduction

The rapid growth of the Internet results in unprecedented number and scale of security threats. Intrusion detection systems (IDS) [1,3] have been developed in turn to protect the availability, confidentiality and integrity of the network systems [3] and prevent unauthorised use of system resources. Both theoretical models of intrusion detection and reaction [4] and real systems such as SNORT, classified as intrusion prevention/detection system (IPS/IDS) [18] are developed. As the nature of threats dynamically evolves, statistical and artificial intelligence techniques are applied to develop adaptive intrusion detection systems. Such systems are developed at two levels, namely host (Host-Based IDS) [5] and network level (Network IDS, NIDS) [16,20].

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 68–77, 2012.

Large amounts of data and significant number of threats processed by NIDS resulted in the extensive use of artificial intelligence (AI) techniques in this field. Among them hybrid artificial intelligence systems (HAIS) have been shown to be of particular use. In particular, in [10] a combination of genetic algorithms and fuzzy systems was used to optimise the layout of distributed network sensors used to monitor network traffic. A related idea of a hybrid system combining statistical methods and Petri networks to process data from detection sensors is proposed in [2]. Another approach is based on the idea of combining conventional intrusion detection methods with AI techniques. Among other, the solution proposed in [6] combines immune-inspired methods with conventional network intrusion detection and monitoring methods.

As far as Network Intrusion Detection Systems are concerned, one of the main issues to address is the constantly growing volume of data transferred in modern networks. In the case of TCP connections, each connection is precisely defined by a sequence of datagrams. Formally, the sequence of datagrams of a connection could be represented by a vector x. However, it would be virtually impossible to develop some of the models, such as multilayer perceptrons (MLPs) [9] with thousands or even millions of inputs. Hence, numerous methods of network traffic representation were proposed. Among these methods, the aggregate features of logical network flows, e.g. of TCP connections were proposed. These include the set of 249 discriminators proposed by A. Moore et al. in [17]. The discriminators include a number of features calculated based on the actual sequence of datagrams contained in the x vector, such as maximum of bytes in (Ethernet) packet [17]. From the data processing point of view, the variable dimensionality of the raw traffic data is replaced with a globally constant dimensionality of the flow vectors describing every flow in the network, including TCP and UDP, but also other protocols. It has been shown that some data processing tasks can be successfully performed based on the flow records defined in this way. Among others, H. Kim et al. [12] experimentally showed that flow features-based classification can be successfully used to categorise traffic by service (e.g. http, ftp, smtp).

While flow feature records can be used as an input for data mining models, the impact of individual features on misuse detection remains largely unclear. Hence, in IDS, both feature selection [12,11,20] and dimensionality reduction (DR) [15,22,21] are applied to limit the number of input signals fed to the models. In the case of flow-based detection, the number of flow features can exceed 200, in case most features proposed by A. Moore are applied. Hence, the need for reducing the number of model inputs is also observed.

At the same time, the scale of dimensionality reduction of features proposed in [17] or used in [12] is not discussed in these works and remains largely an open issue. In the case of prediction models, it has been observed that the scale of dimensionality reduction should be controlled by the quality of prediction models developed with the data. In particular, it was shown that a priori data reduction may result in suboptimal prediction models [8]. This study proposes a hybrid AI method combining conventional methods with artificial intelligence

techniques. More precisely, the process of dimensionality reduction is combined with neural networks and decision trees and controlled by the accuracy of misuse detection. A method controlling the scale of dimensionality reduction in view of the accuracy of classification models built with the data transformed by DR is proposed. Simulation results show that the accuracy of the proposed method exceeds the accuracy attained when standard DR techniques, such as widely used [13,14,15] scree plot or proportion of explained variation, are applied to control DR process.

One more outstanding issue is the reduction of false positive (FP) errors [4,1] and the evaluation of the impact of DR on FP rates. Therefore, the objective of this study is to reconsider the existing techniques of selecting reduced dimension of the data. This should be done in view of the compromise between minimal overall and FP classification rates, and maintaining possibly minimal set of input features supplied to the classification model. This formulates a difficult multiobjective optimisation problem this study is aiming to contribute to.

One of the factors hindering the research on Internet vulnerabilities is lack of realistic and representative data sets [7]. To address the objectives formulated above, our works relies on the fundamental work of M. Žádník and Z. Michlovský [19]. The authors developed the real network flow data set to analyse one of the key vulnerabilities of Internet systems i.e. spam messages. The data set made available by the authors, is discussed in detailed in Sect. 2. The flow records labelled with spam category enable research into the accuracy of net flow-based misuse detection.

In order to analyse the impact of DR on network flow classification, hybrid classifiers based on the combination of DR, decision trees and MLPs were developed. The impact of reduced dimensions on the spam classifiers is evaluated through the extensive battery of simulations.

The remainder of this paper is organised as follows:

- Sect. 2 presents the problem and the data set used in the experiments,
- Sect. 3 discusses the proposed methods and the way they are used to deal with network flows and spam detection,
- Next, results are discussed in Sect. 4, which is followed by the conclusions outlined in Sect. 5.

2 Flow Features-Based Spam Detection

2.1 The Spam Data Set

This study follows the fundamental work of M. Žádník and Z. Michlovský [19]. The authors captured network traces for the communication with the mail server, created the net flow records from the traces and labelled every flow record with an appropriate class. The delivered mails were classified by SpamAssassin into two groups: relevant emails and spam [19]. The details of the procedure can be found in [19]. The data set contains 58 042 records, with 64 non-constant input features each. The number of records in every class is different, which is

Table 1. Network flow categories

Class name	No. of records	Class description
y_spam	11222	Spam messages, as classified by SpamAssassin
n_spam	1554	Valid messages
dnsbl	38314	Spam messages, rejected based on server black list
relay	2618	Outgoing messages, sent from the monitored mail server
other	4334	Traffic caused by scanning, DoS, etc.

summarised in Table 1 [19]. The results of the work and the data set developed by the authors were used as a starting point for this study.

2.2 Data Preprocessing

To accommodate the training process, a more balanced data set can be created. Due to relatively high number of records in every class, a decision was made to first apply undersampling to the original data set. Therefore, the number of records in every class, was reduced by random selection to $card(\{x : class(x) = n\_spam\})$ i.e. to 1554. Moreover, from the network point of view, it is important to classify network flows into *spam* flows and *correct* flows, irrespective of the subcategories of this division. Finally, the third class meaning *other* can be used to contain other traffic such as traffic caused by scanning. Hence, a decision was made to reduce the number of classes. In particular, one *spam* class contains both *dnsbl* and *y_spam* original classes. Similarly the *correct* class contains both *relay* and *n_spam* classes. For the purpose of the remaining part of this study, the three classes: *spam*, *correct* and *other* are used to mark all the patterns.

3 Model Development and Data Reduction

3.1 Key Objectives

The data set described above provides basis for the construction of classifier models. Moreover, under laboratory conditions, flow records of arbitrary sizes can be considered and processed efficiently. However, in modern telecom networks, the observed network traffic can exceed 10Gbit/s. This makes the need for high performance data collection and investigation even more significant than before. Hence, the error rates of a classification model, should be analysed also in terms of the computing overhead of both the model and the data capture module producing flow records out of raw network data. It should be emphasised that both operations i.e. flow record production and flow classification should be performed in near real-time conditions. Hence, in the analysed case of network flow-based classification, an important aspect is to minimise the length of a flow record required to classify every flow. The impact of this reduction on the classification accuracy might be different, depending on the category of classification

model. Hence, one of the objectives of model development and simulation scenarios is to experimentally evaluate the impact of flow data reduction on different classification models.

3.2 The Evaluation Method

The analysis of high-dimensional data aims to identify and eliminate redundancies among the observed variables [15]. The process of DR is expected to reveal underlying latent variables. More formally, the DR of a data set $D \subset \mathbb{R}^S$ can be defined by a function used to code an element $x \in D$ [15]:

$$c : \mathbb{R}^S \longrightarrow \mathbb{R}^R,\ x \longrightarrow \tilde{x} = c(x) \tag{1}$$

In the analysed case of the dimensionality reduction of flow records D the primary objective is to minimise the reduced record length R, while maintaining possibly high classification accuracy and relatively limited false positive error rates. To objectively investigate the impact of the reduction on the spam detection classifier, Alg. 1 was used. The algorithm is prepared to work with the data sets composed of a relatively limited number of records. More precisely, Cross-Validation (CV) is used in $EvaluateDimension()$ procedure to evaluate the merits of the data transformation.

Input: D - matrix of input attributes, $P \subset \mathbb{R}$ - vector of corresponding output features, $card(D)$=$card(P)$, K - the number of CV folders, r - the number of training sessions
begin
  for $R = 2, \ldots, S$ **do**
    $[E^R_{\mathrm{V}}(D,P), F^R_{\mathrm{V}}(D,P), E^R_{\mathrm{T}}(D,P), F^R_{\mathrm{T}}(D,P)] =$ EvaluateDimension(D,P,K,R,r);
  end
end

Algorithm 1. The evaluation of the impact of dimensionality reduction on classifier accuracy

The core of the proposed method is the evaluation of the impact of dimensionality reduction to dimension R on the classification model built with $c(D) \subset \mathbb{R}^R$. This evaluation is described by Alg. 2 and considers different divisions of the available data set and several runs of model construction algorithm to minimise the impact of individual sessions on dimension evaluation. Moreover, both $E()$ rate standing for classification error and $F()$ standing for false positive rates are calculated. In the analysed case, by false positive a classification of a non-spam flow i.e. *correct* or *other* flow, to *spam* class is considered. The algorithms automate the task of evaluating the impact of dimensionality reduction method defined by $DimensionalityMapping()$ on the classification models.

Input: D - matrix of input attributes, $P \subset \mathbb{R}$ - vector of corresponding output features, $card(D)$=$card(P)$, K - the number of CV folders, R - reduced dimension, r - the number of training sessions
Data: D_i - a family of K sets: $D_i \cap D_j = \emptyset, i \neq j$, $\cup_{i=1}^{K} D_i = D$; $P_i, P_{\mathrm{L}}, P_{\mathrm{T}}, P_{\mathrm{V}}$ - output features corresp. to $D_i, D_{\mathrm{L}}, D_{\mathrm{T}}, D_{\mathrm{V}}$
Result: $E_{\mathrm{T}}^{R}(D,P)$ - the average classification error rate of the models on the testing sets; $E_{\mathrm{V}}^{R}(D,P)$ - the average classification error rate of the models on the validation sets; $F_{\mathrm{T}}^{R}(D,P)$ - the average false positive error rate of the models on the testing sets; $F_{\mathrm{V}}^{R}(D,P)$ - the average false positive error rate of the models on the validation sets.

begin
  c = DimensionalityMapping(D,R);
  for $i = 1, \ldots, r$ **do**
    $\{D_j, P_j\}_{j=1,\ldots,K}$ = DivideSet(D,P,K);
    for $k = 1 \ldots K$ **do**
      $D_{\mathrm{T}} = D_k$;
      $D_{\mathrm{V}} = D_{(k+1) \bmod K}$;
      $D_{\mathrm{L}} = \cup_{j \in \{1,\ldots,K\}-\{k\}-\{(k+1) \bmod K\}} D_j$;
      $\tilde{D}_{\mathrm{L}}^{R} = c(D_{\mathrm{L}})$;
      $\tilde{D}_{\mathrm{T}}^{R} = c(D_{\mathrm{T}})$;
      $\tilde{D}_{\mathrm{V}}^{R} = c(D_{\mathrm{V}})$;
      $M = train(\tilde{D}_{\mathrm{L}}^{R}, P_{\mathrm{L}}, \tilde{D}_{\mathrm{V}}^{R}, P_{\mathrm{V}})$;
      $E_{\mathrm{T}}((i-1)*K+k) = E(M(\tilde{D}_{\mathrm{T}}^{R}), P_{\mathrm{T}})$;
      $E_{\mathrm{V}}((i-1)*K+k) = E(M(\tilde{D}_{\mathrm{V}}^{R}), P_{\mathrm{V}})$;
      $F_{\mathrm{T}}((i-1)*K+k) = F(M(\tilde{D}_{\mathrm{T}}^{R}), P_{\mathrm{T}})$;
      $F_{\mathrm{V}}((i-1)*K+k) = F(M(\tilde{D}_{\mathrm{V}}^{R}), P_{\mathrm{V}})$;
    end
  end
  $E_{\mathrm{T}}^{R}(D,P) = median(E_{\mathrm{T}}())$;
  $E_{\mathrm{V}}^{R}(D,P) = median(E_{\mathrm{V}}())$;
  $F_{\mathrm{T}}^{R}(D,P) = median(F_{\mathrm{T}}())$;
  $F_{\mathrm{V}}^{R}(D,P) = median(F_{\mathrm{V}}())$;
end

Algorithm 2. EvaluateDimension() procedure

4 Experimental Results

The experiments performed in this study involved decision trees and multilayer perceptrons as classification models. Separate runs of Alg. 1 were performed for these two types of models. In both cases, the *train*() method being a part of Alg. 2 was used to train a model being a decision tree or an MLP network using training $(\tilde{D}_{\mathrm{L}}^{R}, P_{\mathrm{L}})$ and validation $(\tilde{D}_{\mathrm{V}}^{R}, P_{\mathrm{V}})$ data sets. Moreover, it was aiming to

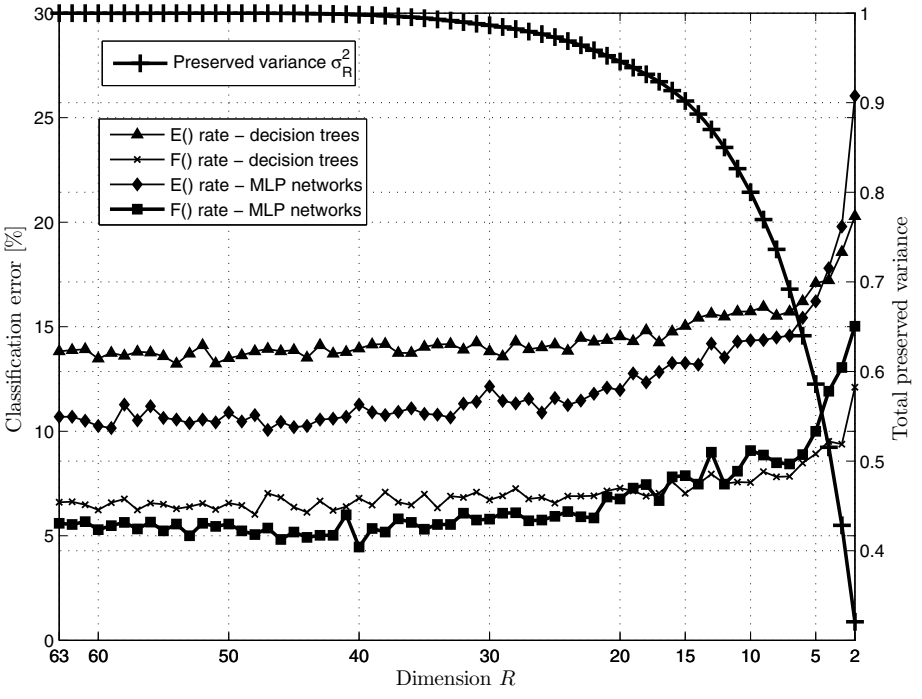


Fig. 1. Spam detection error rates as a function of reduced dimension R

improve the generalisation capabilities of a model. In the case of decision tress, first the construction of an excessive tree with minimum split criterion set to 1 record was done. Then, the tree was reduced by eliminating unnecessary leaves and divisions, in order to minimise the classification error rate measured on the validation data set. In the case of MLP networks, early stopping based on the minimisation of the validation error was applied to avoid overtraining.

As far as dimensionality reduction is concerned, one of the most popular techniques, being Principal Component Analysis (PCA) [15] was used. Hence, the role of $DimensionalityMapping()$ was to return an $R \times card(D)$ PCA transformation matrix that was used next to reduce the dimensionality of the original data.

Let ϱ denote a $(S \times S)$ covariance matrix, $\lambda_1, \ldots, \lambda_S$ denote eigenvalues of the matrix sorted in descending order and $\mathbf{q}_1, \ldots, \mathbf{q}_S$ the associated eigenvectors i.e. $\varrho \mathbf{q}_i = \lambda_i \mathbf{q}_i, i = 1, 2, \ldots, S$. Finally, let $\mathbf{a} : a_j = \mathbf{q}_j^T \mathbf{x} = \mathbf{x}^T \mathbf{q}_j,\ j = 1, \ldots, S$. The coding function $c^R(\mathbf{x})$ reducing dimension to R was defined as $c^R(\mathbf{x}) = [a_1, \ldots, a_R]$.

Extensive simulations were performed to analyse the error rates $E()$ and $F()$ over a range of reduced dimensions $R \in \{2, ..., 63\}$, with $K = 3$ and $r = 4$. In the case of MLP networks, 10 hidden neurons were used. The results of the experiments are summarised in Fig. 1. The error rates on the testing data sets

Table 2. The summary of the search for the optimal dimension R

		E() - overall classification		F() - false positive rate	
R_{SP}	R_{PEV}	R_{MLP}	R_{DT}	R_{MLP}	R_{DT}
35	21	30-33	18-20	22-24	15-18

calculated by Alg. 1 are shown together with the Proportion of Explained Variation (PEV) value corresponding to the analysed dimension R.

As both decision tree creation and the training of MLP networks is partly affected by different divisions of the entire data set and the nature of the model construction process, which may produce different models, the resulting $E()$ and $F()$ curves are partly noisy. Nevertheless, the following conclusions can be made:

- The flow records can be largely reduced from the original $S = 64$ dimensions. A reduction to ca. 20 signals can largely preserve the error rate levels observed at $R = 63$.
- Should higher data reduction be needed due to performance reasons, this may impact the model selection. More precisely, MLP networks outperform decision trees for $R \geq 5$, when overall rate $E()$ is considered, while the opposite tendency is observed for $R < 5$.
- The selection of the best classification model depends on the category of error rate to minimise. In case, the minimisation of false positive errors is the dominating criterion and $5 < R < 15$, decision trees provide lower error rates. For the same dimensionality of the data, MLP networks provide lower overall classification error.

Diverse values of optimal reduced dimension R selected by different techniques are summarised in Table 2. R_{SP} and R_{PEV} stand for the reduced dimension as suggested by scree plot analysis and PEV criterion. In the latter case $R_{\mathrm{PEV}} = min_{l=1,2,\ldots,S} : \frac{\sum_{d=1}^{l} \lambda_d}{\sum_{d=1}^{S} \lambda_d} \geq 0.95$. The remaining values R_{MLP} and R_{DT} show the optimal dimensions when a compromise between the minimisation of $E()$ and $F()$ and the minimisation of R is sought, while using MLP networks and decision trees, respectively. What should be emphasised is that the a priori selection of reduced dimension $R \in \{R_{\mathrm{SP}}, R_{\mathrm{PEV}}\}$ may result in suboptimal R values. Equally importantly, the optimal dimension is largely different depending on the classification model. For MLP networks, when overall classification accuracy is the dominating criterion, $R \in \{30, \ldots, 33\}$ should be preferred, while the use of DT results in $R \in \{18, 19, 20\}$. Moreover, in both cases the suggested values are not the values resulting from scree plot analysis.

5 Summary

A method controlling the selection of reduced dimension in view of large scale network data processing was proposed. It was shown that the optimal reduced

dimension depends not only on the a priori investigation of input matrix, but also on the classifier used and the error rate to be minimised. The experiments show that it is possible to classify network flows with relatively high accuracy by using a very limited number of features. What should be emphasised is that when the reduction of false positives is an issue, the scale of dimensionality reduction might be different comparing to the way the same process should be performed when an overall error rate is an issue. Taking into account the excessive volume of data that has to be constantly analysed for possible misuse of the internet services, extensive experiments aiming at selecting the appropriate reduction of the data are fully justified.

Future works should concentrate on the reduction of false positive errors and the reduction of computational cost of the data transformation by eliminating the input attributes having a minor impact on the reduced features. Moreover, other dimensionality reduction techniques are planned to be included in the proposed framework.

Acknowledgments. We would like to thank R. Filasiak for his suggestions and M. Žádník for sharing his experience and the data set used to develop the models discussed in this study.

References

1. Abouabdalla, O., et al.: False Positive Reduction in Intrusion Detection System: A Survey. In: Proc. of IC-BNM 2009, pp. 463–466 (2009)
2. Baláž, A., Trelová, J., Kostráb, M.: Architecture of Distributed Intrusion Detection System Based on Anomalies. In: 14th International Conference on Intelligent Engineering Systems (INES), pp. 79–83 (2010)
3. Barapatre, P., et al.: Training MLP Neural Network to Reduce False Alerts in IDS. In: Proc. of the 2008 Int. Conf. on Computing, Communication and Networking, ICCCN 2008 (2008)
4. Biskup, J.: Security in Computing Systems. Challenges, Approaches and Solutions. Springer, Heidelberg (2009)
5. Dash, S.K., Rawat, S., Pujari, A.K.: Use of Dimensionality Reduction for Intrusion Detection. In: McDaniel, P., Gupta, S.K. (eds.) ICISS 2007. LNCS, vol. 4812, pp. 306–320. Springer, Heidelberg (2007)
6. Fanelli, R.L.: A Hybrid Model for Immune Inspired Network Intrusion Detection. In: Bentley, P.J., Lee, D., Jung, S. (eds.) ICARIS 2008. LNCS, vol. 5132, pp. 107–118. Springer, Heidelberg (2008)
7. Fomenkov, M., Claffy, K.: Internet measurement data management challenges. In: The Cooperative Association for Internet Data Analysis (CAIDA), San Diego, USA (2011)
8. Grzenda, M.: Prediction-Oriented Dimensionality Reduction of Industrial Data Sets. In: Mehrotra, K.G., Mohan, C.K., Oh, J.C., Varshney, P.K., Ali, M. (eds.) IEA/AIE 2011, Part I. LNCS (LNAI), vol. 6703, pp. 232–241. Springer, Heidelberg (2011)
9. Haykin, S.: Neural Networks: a Comprehensive Foundation. Prentice-Hall Inc. (1999)

10. Hu, C., et al.: On the Deployment Strategy of Distributed Network Security Sensors. In: 13th IEEE International Conference on Networks (2005)
11. El-Khatib, K.: Impact of Feature Reduction of the Efficiency of Wireless Intrusion Detection Systems. IEEE Trans. on Parallel and Distributed Systems 21(8), 1143–1149 (2010)
12. Kim, H., et al.: Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices. In: Proc. of ACM CoNEXT 2008 (December 2008)
13. Larose, D.T.: Data Mining Methods and Models (2006)
14. Lattin, J.M., Carroll, J.D., Green, P.E.: Analyzing Multivariate Data (2003)
15. Lee, J., Verleysen, M.: Nonlinear Dimensionality Reduction. Springer, Heidelberg (2010)
16. Lim, S.Y., Jones, A.: Network Anomaly Detection System: The State of Art of Network Behaviour Analysis. In: Int. Conf. on Convergence and Hybrid Information Technology, pp. 459–465 (2008)
17. Moore, A., Zuev, D., Crogan, M.: Discriminators for use in flow-based classification. Technical Report, RR-05-13, Department of Computer Science, Queen Mary, University of London (2005)
18. http://www.snort.org/
19. Žádník, M., Michlovský, Z.: Is Spam Visible in Flow-Level Statistics? CESNET National Research and Education Network, Prague, Czech Republic, Technical Report 6/2008, 67–78 (2008)
20. Zhang, J., Zulkernine, M., Haque, A.: Random-Forests-Based Network Intrusion Detection Systems. IEEE Trans. on Systems, Man, and Cybernetics 38(5), 649–659 (2008)
21. Zhou, Y.-P.: Hybrid Model Based on Artificial Immune System and PCA Neural Networks for Intrusion Detection. In: Proc. of 2009 Asia-Pacific Conf. on Information Processing, pp. 21–24 (2009)
22. Yanwei, F., Yingying, Z., Haiyang, Y.: Study of Neural Network Technologies in Intrusion Detection Systems. In: Proc. of the 5th Int. Conf. on Wireless Communications, Networking and Mobile Computing (2009)

Encrypting Digital Images Using Cellular Automata

A. Martín del Rey[1], G. Rodríguez Sánchez[2], and A. de la Villa Cuenca[3]

[1] Department of Applied Mathematics
E.P.S. de Ávila, Universidad de Salamanca
C/Hornos Caleros 50, 05003-Ávila, Spain
delrey@usal.es
[2] Department of Applied Mathematics
E.P.S. de Zamora, Universidad de Salamanca
Avda. Cardenal Cisneros 34, 49022-Zamora, Spain
gerardo@usal.es
[3] Department of Applied Mathematics and Computation, E.T.S.I. (ICAI)
Universidad Pontificia Comillas
C/Alberto Aguilera 23, 28015-Madrid, Spain
avilla@upco.es

Abstract. In this paper a novel symmetric protocol to cipher digital images is introduced. The protocol proposed consists of two iterative phases: The confusion phase and the diffusion phase. The first stage permutes the pixels in the image using a discrete chaotic map (specifically, the Cat map), whereas in the second stage, the pixel values (that is, the color of each pixel) are modified sequentially by means of a reversible memory cellular automata. The proposed protocol is shown to be secure against the more important cryptanalytic attacks.

Keywords: Cryptography. Cellular automata. Image processing.

1 Introduction

With the advent of personal computers and the Internet, huge amount of digital visual data are stored on different media and exchanged over various sorts of open networks nowadays. Usually, these visual data contain confidential or private informations. As a consequence, and taking into account this new environment, there are several security problems associated with the processing and transmission of digital images: It is necessary to assure the confidentiality, the integrity and the authenticity of the transmitted digital image. Due to the large amount of data involved in the transmission of digital images only symmetric encryption protocols can be used. To meet these challenges, a wide variety of cryptographic protocols have appeared in the scientific literature (see, for example [1,2]). Traditional data cryptosystems exhibits some drawbacks and weakness in the encryption of digital images (for example, low-level efficiency when the image is large); consequently, they are not suitable for image encryption. In this

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 78–88, 2012.

respect, two-dimensional chaotic maps are naturally employed as each digital image can be represented as a two-dimensional array of pixels ([5,6,7,13]). Moreover, chaos-based and dynamical systems-based algorithms have shown their superior performance: They have many important properties such as the sensitivity dependence on initial conditions and system parameters, pseudorandom properties, ergodicity, nonperiodicity and topological transitivity. Most properties meet some requirements such as sensitive to keys, diffusion and mixing in the sense of cryptography.

In [5], J. Fridrich suggested that image encryption protocols based on chaotic maps should compose of two iterative stages: Chaotic confusion stage and pixel diffusion stage. The confusion stage permutes the pixels of the image without changing its value (the color of the pixel) by using an adequate two-dimensional chaotic map such as Baker map, Cat map or the Standard chaotic map. In this phase, the parameters of the chaotic map serve as the confusion key. In the diffusion stage, the pixel values are modified sequentially such that a small change in the value of only one pixel is spread out to many pixels (avalanche effect). The initial value or the control parameter of the diffusion function serves as the diffusion key. To decorrelate the relationship between adjacent pixels, there must be $n \geq 1$ permutation rounds in the confusion stage. The whole confusion-diffusion round repeats for a number of times to achieve a satisfactory level of security.

The main goal of this paper is to introduce a new symmetric image encryption protocol following the paradigm stated by Fridrich at 1998. In this work, the confusion stage is carryied out by using the Cat map which exhibits good cryptographic properties (see [8]), and in the diffusion stage the use of a suitable reversible memory cellular automata with good diffusion properties is proposed.

Cellular automata are finite state machines formed by a collection of n memory units called cells. At each time step, they are endowed with a state from the state set given by a finite field (see, for example, [10,12]). The state of a particular cell is updated synchronously according to a specified rule function, whose variables are the states of the neighbor cells at the previous time step. Finite state machines (and, consequently, cellular automata) are ubiquitous in modeling the behavior of single threaded embedded systems, in such a way that they capture the states that computer-based systems incarnate and the transitions they perform. Finite state machines have been used in many software engineering methodologies, tools, and developing approaches and continue to be core to widely supported software modeling standards. In this sense, several applications to cryptography have been appeared in the last years (see, for example, [9]).

The rest of the paper is organized as follows: In section 2, the basic definitions and results about the Cat map and cellular automata are introduced; the encryption scheme is presented in section 3 and its security analysis is shown in section 4. Finally, the conclusions are introduced in section 5.

2 Mathematical Preliminaries

2.1 Chaotic Discrete Maps

Arnold's discrete Cat map is a simple discrete dynamical system that stretches and "folds" the trajectories in phase space which is a typical feature of chaotic processes. Specifically, the Cat map is the best known example of Anosov diffeomorphism; it is a two-dimensional invertible chaotic map given by the following transformation (see, for example [3]):

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix} \cdot \begin{pmatrix} x_n \\ y_n \end{pmatrix} (\mathrm{mod}\, 1)\,, \tag{1}$$

where a, b are control parameters, and the notation $x\,(\mathrm{mod}\,1)$ stands for the fractional parts of a real number x by subtracting or adding an appropiate integer number. The Cat map is non-Hamiltonian, nonanalytic and mixing. As the determinant of its linear transformation matrix is equal to 1, it is also area-preserving. Moreover, its Lyapunov characteristic exponents are $\sigma_1 = 2.61803 > 1$ and $\sigma_2 = 0.381966 < 1$, and as the leading Lyapunov characteristic exponent is strictly larger than 1, then the Cat map is chaotic.

2.2 Cellular Automata

Two-dimensional cellular automata (CA for short) are finite state machines given by a 4-uplet $\mathcal{A} = (C, S, V, f)$, where $C = \{\langle i, j\rangle \quad 1 \leq i \leq r, 1 \leq j \leq c\}$ is the cellular space formed by a rectangular array of $r \times c$ memory units called cells which, at every step of time, are endowed with a state from the state set $S = \mathbb{Z}_2 = \{0, 1\}$. The neighborhood of the cell is defined by the finite set of indexes $V \subset \mathbb{Z} \times \mathbb{Z}$, such that for the cell $\langle i, j\rangle \in C$ its neighborhood is:

$$V_{ij} = \{\langle i + \alpha, j + \beta\rangle, (\alpha, \beta) \in V\}. \tag{2}$$

In this work, Moore neighborhood is considered: It is formed by the eight nearest cells around a cell and itself, that is:

$$V = \{(-1,-1)\,, (-1,0)\,, (-1,1)\,, (0,-1)\,, (0,0)\,, (0,1)\,, (1,-1)\,, (1,0)\,, (1,1)\} \tag{3}$$

Moreover, the local transition function $f\colon \mathbb{Z}_2^9 \to \mathbb{Z}_2$ is the function determining the evolution of the CA throughout the time, *i.e.*, the changes of the states of every cell taking the states of its neighbors into account. Hence, if $s_{ij}^t \in \mathbb{Z}_2$ stands for the state of the cell $\langle i, j\rangle$ at time t, the next state of such cell is given by the formula:

$$s_{ij}^{t+1} = f\left(s_{i-1,j-1}^t, s_{i-1,j}^t, s_{i-1,j+1}^t, s_{i,j-1}^t, s_{ij}^t, s_{i,j+1}^t, s_{i+1,j-1}^t, s_{i+1,j}^t, s_{i+1,j+1}^t\right). \tag{4}$$

As the cellular space is finite, periodic boundary conditions must be established in order to assure that the evolution of the cellular automata is well-defined:

$$s_{ij}^t = s_{uv}^t \Leftrightarrow i \equiv u\,(\mathrm{mod}\, r) \text{ and } j \equiv v\,(\mathrm{mod}\, c)\,. \tag{5}$$

Note that the state of each cell at a particular time step t depends only on the states of its neighbor cells at the previous time step $t-1$. Nevertheless, when this dependence is extended to the states of the neighbor cells at previous time steps $t-2, t-3, \ldots$, the CA is called memory CA. Moreover, a CA is called reversible when there exists another cellular automaton which produces the inverse evolution ([11]).

The set of states of all cells at time t is called the configuration at time t and it is represented by the matrix:

$$C^t = \begin{pmatrix} s_{11}^t & s_{12}^t & \cdots & s_{1c}^t \\ s_{21}^t & s_{22}^t & \cdots & s_{2c}^t \\ \vdots & \vdots & \ddots & \vdots \\ s_{r1}^t & s_{r2}^t & \cdots & s_{rc}^t \end{pmatrix}. \tag{6}$$

If we denote by $\mathcal{C}$ the set of all possible configurations of a CA, the global transition function of the cellular automata is a transformation that yields the configuration at the next time step during the evolution of the CA. That is:

$$\begin{aligned} \Phi &: \mathcal{C} \to \mathcal{C} \\ C^t &\mapsto C^{t+1} = \Phi\left(C^t\right) \end{aligned} \tag{7}$$

or

$$\begin{aligned} \Phi &: \mathcal{C} \times \ldots \times \mathcal{C} \to \mathcal{C} \\ \left(C^t, \ldots, C^{t-k}\right) &\mapsto C^{t+1} = \Phi\left(C^t, \ldots, C^{t-k}\right) \end{aligned} \tag{8}$$

for memory CA.

This work deals with reversible memory cellular automata of the form:

$$C^{t+1} = \Phi\left(C^t, \ldots, C^{t-7}\right) = \bigoplus_{k=0}^{6} \Psi_k\left(C^{t-k}\right) \oplus C^{t-7}, \tag{9}$$

where Ψ_k is defined by the following local transition function:

$$\begin{aligned} s_{ij}^{t+1} = {} & \lambda_1^k s_{i-1,j-1}^t \oplus \lambda_2^k s_{i-1,j}^t \oplus \lambda_3^k s_{i-1,j+1}^t \oplus \lambda_4^k s_{i,j-1}^t \oplus \lambda_5^k s_{i,j}^t \\ & \oplus \lambda_6^k s_{i,j+1}^t \oplus \lambda_7^k s_{i+1,j-1}^t \oplus \lambda_8^k s_{i+1,j}^t \oplus \lambda_9^k s_{i+1,j+1}^t, \end{aligned} \tag{10}$$

where $\lambda_i^k \in \mathbb{Z}_2$ for every $1 \leq i \leq 9$. This type of cellular automata is reversible and it was introduced by Fredkin (see [4]). Its evolution backwards is possible by means of the following inverse cellular automata:

$$C^{t+1} = -\bigoplus_{k=0}^{6} \Psi_{6-k}\left(C^{t-k}\right) \oplus C^{t-7}. \tag{11}$$

3 The Encryption Scheme

Set a gray-level digital image defined by $N \times N$ pixels such that p_{ij} stands for the numeric value of the color of the (i,j)-th pixel. Let $I = (p_{ij})$ be the matrix defining the digital image. As is mentioned in the Introduction, the cryptographic algorithm proposed in this work consists of two iterative stages: the confusion and the diffusion phases.

The confusion phase is carryied out using a discrete chaotic map and the number of iterations is T_0. In the diffusion phase a reversible memory CA is used and the configuration of order T_1 is computed.

The Confusion Phase. In this phase, the pixels of the image will be permuted using the discretized Cat map over the image lattice:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix} \cdot \begin{pmatrix} x_n \\ y_n \end{pmatrix} (\mathrm{mod}\, N)\,. \tag{12}$$

The confusion key is formed by the control parameters a and b, and the number of iterations T_0. The matrix associated to the permuted image is J.

The Diffusion Phase. In this phase the value of the pixels of the permuted image will be changed according to a reversible memory cellular automata defined by (9)-(10) where $\lambda_i^k \in \mathbb{Z}_2$ for every $1 \leq i \leq 9$ are random bits standing for the secret key K.

Let $J = (q_{ij})$ be the $N \times N$ matrix obtained from the confusion phase. where $q_{ij} = \left(q_{ij}^1, q_{ij}^2, \ldots, q_{ij}^8\right) \in \mathbb{Z}_2^8$, with $1 \leq i,j \leq N$ is the binary representation of $0 \leq q_{ij} \leq 255$. As a consequence, and taking into account the bits forming the binary representation of the numerical value of each pixel, eight binary matrices with coeffients in $\mathbb{Z}_2$ can be extracted from J:

$$J_k = \begin{pmatrix} q_{11}^k & q_{12}^k & \cdots & q_{1N}^k \\ q_{21}^k & q_{22}^k & \cdots & q_{2N}^k \\ \vdots & \vdots & \ddots & \vdots \\ q_{N1}^k & q_{N2}^k & \cdots & q_{NN}^k \end{pmatrix}, \quad 1 \leq k \leq 8. \tag{13}$$

These matrices are interpreted as the initial configurations of the memory CA defined by (9), that is:

$$C^0 = J_1, C^1 = J_2, \ldots, C^7 = J_8. \tag{14}$$

Then the T_1-th order configutation is computed by means of the last mentioned memory CA:

$$\begin{aligned} C^8 &= \bigoplus_{k=0}^{6} \Psi_k \left(C^{7-k} \right) \oplus C^0 \\ C^9 &= \bigoplus_{k=0}^{6} \Psi_k \left(C^{8-k} \right) \oplus C^1 \\ &\cdots \\ C^{T_1-7} &= \bigoplus_{k=0}^{6} \Psi_k \left(C^{T_1-8-k} \right) \oplus C^{T_1-15} \\ &\cdots \\ C^{T_1} &= \bigoplus_{k=0}^{6} \Psi_k \left(C^{T_1-1-k} \right) \oplus C^{T_1-8} \end{aligned} \tag{15}$$

The matrix representing the ciphered image, $\tilde{J}$, is reconstructed using the eight binary matrices $C^{T_1-7}, \ldots, C^{T_1}$, that is,

$$\tilde{J} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{pmatrix}, \tag{16}$$

where

$$w_{ij} = \left(s_{ij}^{T_1-7}, s_{ij}^{T_1-6}, \ldots, s_{ij}^{T_1} \right), \quad 1 \leq i, j \leq N. \tag{17}$$

The diffusion key is formed by T_1 and $K = \{\lambda_1^1, \ldots, \lambda_9^7\}$.

Note that the secret parameters defining the key of the whole confusion-diffusion phase are a, b, T_0, T_1, K. As a consequence the bit length of the secret key of the algorithm is 128 bits.

3.1 The Decryption Protocol

The decryption of a cipher image $\tilde{J}$ is as follows: Let $\tilde{C}^0, \ldots, \tilde{C}^7$ be the binary matrices extracted from $\tilde{J}$; they stand for the initial configurations of the inverse CA defined by (11). Then, by means of such CA, the configurations $\tilde{C}^{T_1-7}, \ldots, \tilde{C}^{T_1}$ are computed and they yield the image J. Finally, using the suitable number of iterations of the Cat map, the original image I is obtained.

3.2 Computational Complexity

The number of bit operations involving the encryption phase and the decryption phase are the same. Specifically, the confusion phase takes $4T_0N^3$ bit operations whereas the diffusion phase involves $126\,(T_1 - 8)\,N^2$ bit operations. Moreover,

the numerical value of encryption/decryption times for the standard size image of Lena (see Figure 1-(a)) with a standard computer (2.2GHz quad-core Intel Core i7 processor with 6MB shared L3 cache) is 0.000016 seconds for the confusion phase and 0.00022 for the diffusion phase, where the parameters values used are given in equation (18).

4 The Security Analysis

In this section the security of the protocol introduced in the last section is analyzed. Specifically, some cryptanalytic attacks are studied: statistical attacks, sensitivity to initial conditions, differential attack and chosen-plainimage attack.

4.1 Brutte-Force Attacks and Statistical Analysis

As the secret key is of bit-length 128, its key space size is $2^{128} \approx 3.4028 \cdot 10^{38}$. Then, it is large enough to prevent exhaustive searching.

We have also performed a statistical analysis in order to prove the confusion and diffusion properties of the proposed protocol, which allows it to strongly resist statistical attacks. Specifically the histograms of original image and the cipher image are checked and the correlation coefficients are computed.

Let us consider the 256 gray-scale image of size 128×128 given in Figure 1-(a). Its histogram is shown in Figure 1-(c). If we compute the cipher image by means of the proposed protocol with the following parameters:

$$a = 1, b = 2, T_0 = 32, T_1 = 64, K = \texttt{f864a0224824bac4c9a90ea317cff1ef}, \tag{18}$$

the encrypted image obtained is shown in Figure 1-(b). Moreover, its histogram is shown in Figure 1-(d). From the figure one can see that the histogram of the ciphered image is fairly uniform and it is significantly different from that of the original image. It demonstrates that the encryption algorithm has covered up all the characters of the plain image and shows good performance of balanced 0-1 ratio and zero correlation.

The following procedure will be carryied out to test the correlation between two adjacent pixels in the original and the cipher image: First of all randomly select 1.000 pairs of two adjacent pixels from the image, and then, calculate the correlation coefficient of each pair by using the following formula:

$$r_{xy} = \frac{cov\,(x,y)}{\sqrt{D\,(x)}\sqrt{D\,(y)}}, \tag{19}$$

where x and y are the grey-scale values of the two adjacent pixels in the image and:

$$\begin{aligned} cov\,(x,y) &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E\,(x)\right)\left(y_i - E\,(y)\right), \\ E\,(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i, \quad D\,(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E\,(x)\right)^2. \end{aligned} \tag{20}$$

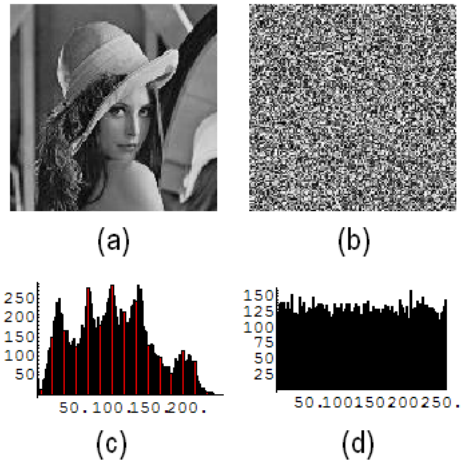


Fig. 1. (a) Lena's picture defined by 128×128 pixels and 256 grey levels; (b) Cipher image of Lena's picture; (c) Histogram of original picture; (d) Histogram of cipher image

As a consequence, the results obtained are shown in Table 1.

Table 1. Correlation coefficients of two adjacent pixels

	Lena picture	Cipher image
Horizontal	0.941845	−0.0534153
Vertical	0.851011	−0.000276273
Diagonal	0.844094	0.0163002

The correlations coefficients of the original image and cipher image are far appart (note that the correlation coefficients of Lena's picture are close to 1, whereas the corresponding coefficients of the cipher image are very close to 0). Consequently, the encryption algorithm satisfy zero co-correlation.

4.2 Sensitivity to Initial Conditions

A desirable property of any cryptographic protocol is that small changes in the secret key should result in a significant change in the ciphered image. For example, if we change the 32-th bit of K in our case, the difference between the two ciphered images is the 99.67 %.

Moreover, if the last trivially modified key is used to decrypt the ciphered image given in Figure 1-(b), the decryption also completely fails as it is shown in Figure 2. As a consequence, it can be concluded that the encryption protocol is sensitive to key, a small change of the key yields a different deciphered image and no information about the original one is obtained.

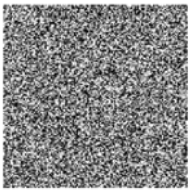


Fig. 2. Deciphered image with the single bit modified key K

4.3 Differential Attack

To test the influence of one-pixel change on the whole ciphered image, two usual measures are used: the number of pixels change rate, $NPCR$, which measures the percentage of different pixel numbers between two images; and the unified average changing intensity, $UACI$, which measures the average intensity of differences between two images.

Suppose that $\tilde{J}_1 = \left(w_{ij}^1\right)$ and $\tilde{J}_2 = \left(w_{ij}^2\right)$ are two ciphered images whose corresponding plainimages differ in only one-pixel. Define a bipolar array of size $N \times N$, $D = (d_{ij})$, such that $d_{ij} = 0$ if $w_{ij}^1 = w_{ij}^2$, and $d_{ij} = 1$ otherwise. The $NPCR$ and $UACI$ are defined as follows:

$$NPCR = \frac{\sum_{i,j=1}^{N} d_{ij}}{N^2} \times 100\%, \, UACI = \frac{1}{N^2} \left(\sum_{i,j=1}^{N} \frac{\mid w_{ij}^1 - w_{ij}^2 \mid}{255} \right) \times 100\%, \quad (21)$$

Some tests have been performed on the proposed scheme about the influence of only one-pixel change on the 256 gray-scale image of size 128×128 given in Figure 1. If we change the $(64, 64)$-th pixel of the original image and its value passes from 29 to 34, the experimentally measured value for $NPCR$ is 86.6857, whereas the value for $UACI$ coefficient is 14.6001. This result indicates that small change in plain image creates significant changes in the ciphered images, so the proposed algorithm is highly resistive against differential attack.

4.4 Other Cryptanalytic Attacks

Finally, we will study the robustness of the protocol against some specific and very selective cryptanalytic attacks which are initially designed for text-based ciphers. Specifically, they are cipher image-only attack, known-plain image attack and chosen-plain image attack. In the cipher image-only attack, an opponent must determine the secret key solely from an intercepted cipher image $\tilde{J}$. In the known-plain image attack, the opponent must deduce the secret key starting from several pairs of plain images and corresponding ciphered images and, finally, in the chosen-plain image, the cryptanalyst is able to choose the plain images and obtain the corresponding cipher images. If the encryption protocol is secure against the chosen-plain image attack, it is also secure against cipher image-only attack and known-plain image attack.

Suppose the opponent chooses a plain image I whose pixels are all of the same color. Then the matrix J obtained from the confusion phase is constant and, consequently, the matrices $J_1, J_2, \ldots, J_8$ (which stand for the configurations $C^0, C^1, \ldots, C^7$ of the evolution of the cellular automata) are binary constant matrices. When the CA is iterated the new configurations obtained are pseudorandom binary matrices since their coefficients are computed by means of XOR sum of the coefficients of J_i perturbed by pseudorandom bits (forming the secret key). Then, the cipher image $\tilde{J}$ is pseudorandom and no information about the secret key is obtained.

5 Conclusions and Further Work

In this paper, a new symmetric encryption protocol for digital images is introduced. It consists of two stages: The confusion stage and the diffusion stage. The confusion stage is given by means of the Cat map, whereas in the diffusion stage, some evolutions of a particular reversible memory cellular automata are computed using as initial configurations eight binary matrices extracting from the original image to be encrypted. The algorithm shows to be secure against the most important cryptanalytic attacks.

Further work is aimed at designing similar encryption protocols involving cellular automata in the confusion stage. Moreover, it is also very interesting the study and classification of the more suitable cellular automata for their use in both stages.

Acknowledgments. This work has been supported by Ministerio de Ciencia e Innovación (Spain) under grants MTM2008-02773 and TIN2011-25452.

References

1. Álvarez Marañón, G., Hernández Encinas, L., Martín del Rey, Á.: A New Secret Sharing Scheme for Images Based on Additive 2-Dimensional Cellular Automata. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005. LNCS, vol. 3522, pp. 411–418. Springer, Heidelberg (2005)
2. Chang, C.-C., Hwang, M.-S., Chen, T.-S.: A new encryption algorithm for image cryptosystems. J. Sist. Software 58, 83–91 (2001)
3. Chen, G., Dong, X.: From chaos to order: methodologies, perspectives and applications. World Scientific, Singapore (1998)
4. Fredkin, E.: Digital mechanics. An informal process based on reversible universal cellular automata. Physica D 45, 254–270 (1990)
5. Fridrich, J.: Symmetric ciphers based on two-dimensional chaotic maps. Int. J. Bifucar. Chaos 8, 1259–1284 (1998)
6. Gao, H., Zhang, Y., Liang, S., Li, D.: A new chaotic algorithm for image encryption. Chaos Soliton Frac. 29, 393–399 (2006)
7. Guan, Z.H., Huang, F.J., Guan, W.J.: Chaos-based image encryption algorithm. Phys. Lett. A 346, 153–157 (2005)
8. Lian, S.G., Sun, J., Wang, Z.: A block cipher based on a suitable use of chaotic standard map. Chaos Soliton Frac. 26, 117–129 (2005)

9. Luengo, I. (ed.): Recent Trends in Cryptography. Contemporary Mathematics, vol. 477. AMS (2009)
10. Toffoli, T., Margolus, N.: Cellular Automata Machines: A New Environment for Modeling. The MIT Press, Cambridge (1987)
11. Toffoli, T., Margolus, N.: Invertible cellular automata: A review. Physica D 45, 229–253 (1990)
12. Wolfram, S.: A New Kind of Science. Wolfram Media, Champaign (2002)
13. Zhang, L.H., Liao, X.F., Wang, X.B.: An image encryption approach based on chaotic maps. Chaos Soliton Frac. 24, 759–765 (2005)

Self-Organizing Maps versus Growing Neural Gas in Detecting Data Outliers for Security Applications

Zorana Banković, David Fraga, Juan Carlos Vallejo, and José M. Moya

ETSI Telecomunicación, Univ. Politécnica de Madrid,
Av. Complutense 30, 28040 Madrid, Spain
{zorana,dfraga,jcvallejo,josem}@die.upm.es

Abstract. Our previous work has demonstrated that clustering-based outlier detection approach offers numerous advantages for detecting attacks in Wireless Sensor Networks, above all adaptability and the possibility to detect unknown attacks. In this work we provide a comparison of Self-organizing maps (SOM) and Growing Neural Gas (GNG) used for this purpose. Our results reveal that GNG is superior to SOM when it comes to the level of presence of anomalous data during the training, as GNG is capable of detecting the attack even with small portion of normal data during the training, while SOM need the majority of the training data to be normal in order to detect it. On the other hand, after both being trained with normal data, SOM performs somewhat better as the attack becomes more aggressive, i.e. it exhibits higher detection rate, although both are capable of detecting the attack in each case.

Keywords: Self-organizing maps, growing neural gas, outliers, wireless sensor networks.

1 Introduction

In our previous work [1] we have demonstrated that clustering-based outlier detection approach offers numerous advantages for detecting attacks in Wireless Sensor Networks (WSN), such as high adaptability, flexibility, possibility to detect unknown attacks, no restrictions on training data, etc. Regarding clustering, the possibilities are to deploy techniques such as *k*-means, *k-NN*, but also the topology-preserving competitive methods, such as Self-organizing maps (SOM) [2], or Growing Neural Gas (GNG) [3].

The topology preserving techniques are very convenient for our application, since one of the main parameters that reveal the presence of outliers is the average distance of a cluster to its closest neighbors. In the case of topology preserving techniques it is very well known which the closest clusters are, thus making the calculation of the mentioned parameter straightforward. Furthermore, the fact that both techniques use exponentially decaying learning rate makes them less susceptible to the issue of poor initialization. As *k*-means suffers from this problem, their advantage to it is obvious. On the other hand, *k-NN* has been discarded from the start due to its high memory

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 89–96, 2012.

consumption also during the testing process, which is unacceptable in resource limited environments such as WSNs.

SOM and GNG mainly differ in the fact that the size of SOM is fixed from the start, while the size of GNG grows during the training. Fixed size can be a limitation, as it is not possible to know the optimal number of clusters from the start. In order to overcome this issue in the case of SOM, we have deployed genetic algorithm (GA) that in essence searches for the optimal clustering. However, GA consumes lots of resources, which limits its application in WSNs. For this reason, we have implemented GNG in order to overcome the issues of both SOM and GA.

The rest of the work is organized as follows. Section 2 provides some basic information on the previous work in WSN security. Section 3 gives more detail to the description of the problem and on the implemented algorithms, while Section 4 provides experimental evaluation. Finally, conclusions are drawn in Section 5.

2 Previous Work

Recently few solutions that deploy machine learning techniques appeared [7], [10]. Among these solutions we can also find a few anomaly based solutions [8], [9], [11] that claim of having the possibility to detect unknown attacks. They uphold the idea that machine learning techniques offer higher level of flexibility and adaptability to the changes of the environment, as retraining can be done automatically.

However, the feature sets they deploy mostly include those features that capture the properties of known attacks, i.e. those that are known to change under the influence of an attacker, or are known to be weak spots. This is their major deficiency, as relying on these features only the known attacks or their variations can be detected. Furthermore, it assumes that an attacker can exploit only the known vulnerabilities, but general experience is that vulnerability is detected after being exploited by an adversary. Some of them assume that the feature sets can be expanded [7], yet this has to be done through a human intervention.

Thus, regarding previous work on WSN security, we can say that the main deficiencies of the known solutions are: the scope of attacks they can detect is limited and their adaptation has to be performed through human interaction. Thus, our aim is to provide a machine learning based solution that does not suffer from these issues, i.e. a solution that would be capable of detecting wide range of attacks, including the previously unseen ones, which would also be adaptable automatically.

3 Problem Definition

In order to provide uninterrupted network operation in WSNs, core network protocols (aggregation, routing and time synchronization) have to be secured. Regarding the attacks on the aggregation protocol, we assume that they demonstrate themselves in skewed aggregated values, which can be the result of either a number of skewed sensed values, or a compromised aggregated node. The assumption is very

reasonable, having in mind that the main objective of these attacks is to provide wrong picture of the observed phenomenon.

On the other hand, in time critical systems it is mandatory to receive information within certain time window. If the attacker manages to introduce delays or desynchronize clock signal in various nodes, the received critical information will not be up to date, which can destabilize the system. Also, if the received information is not up to date, the aggregated value will be skewed, as it will also be out of date. For these reasons, and given the existing redundancy in WSNs, we believe that these attacks can be detected as temporal and/or spatial inconsistencies of sensed values.

Regarding attacks on routing protocols [4], we assume that they will introduce new and different paths than those that have been seen before. Here we have attacks whose main objective is to compromise the routing protocol, and they usually do it by spoofing or altering the data stored in the routing tables of the nodes. Thus, the resulting routing paths will be different from those used in a normal situation. In the case of wormhole for example, two nodes that are not within each other's radio range result in consecutive routing hops in routing paths, which is not possible in a normal situation. From these examples we can see that the assumption about the attacks resulting in routing paths different from those that appear in normal situation is reasonable. Thus, in this case we want to detect temporal inconsistencies in paths used by each node.

3.1 Feature Extraction and Formation of Model

Following the idea of temporal and/or spatial inconsistency in the presence of attackers, we want to provide the model of the data that would capture these properties and allow us to deploy machine learning.

For the case of sensed values, we follow the idea presented in our previous work [1] based on extracted *n*-grams and their frequencies within different time windows. For the purpose of illustration, we will give a short example for a sensor that detects presence. Let the sensor give the following output during the time window of size 20: 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0. If we fix the *n*-gram size on 3, we extract all the sequences of size 3 each time moving one position forward. In this way we can observe the following sequences and the number of their occurrences within the time window: 111 – occurs 6 times, 110 – 2, 100 – 2, 000 – 6, 001 – 1, 011 – 1. Thus, we can assign them the following sequences: 111 – 0.33, 110 – 0.11, 100 – 0.11, 000 – 0.33, 001 – 0.05, 011 – 0.05. In our model, the sequences are the features and their frequencies are the corresponding feature values. Thus, the sum of the feature values is always equal to 1. This characterization is performed in predefined moments of time and takes the established amount of previous data, e.g. we can perform the characterization after every 20 time periods based on previous 40 values.

In a similar fashion, we form features for spatial characterization. The first step is to establish vicinities of nodes that historically have been giving consistent information. In this way, an *n*-gram for spatial characterization in a moment of time is made of the sensor outputs from that very moment. For example, if sensors S1, S2, S3 that belong to the same group each give the following output: 1 1 1 0 during four time

epochs, we characterize them with the following set of *n*-grams (each *n*-gram contains at the first position the value of S1, the value of S2 at the second and the value of S3 at the third at a certain time epoch): 111 – occurs 3 times, 000 – occurs once, thus the feature value of each *n*-gram is: 111 – 0.75, 000 – 0.25, i.e. the frequencies within the observed period of time.

The same principle is followed for characterizing routes that a node has been using to send its sensed data to the sink. Each routing hop adds its ID to the message that is further forwarded, so the sink gets the information about the routing path together with the message. Each sensor has its own model and each feature, i.e. n-gram in the model consists of a predefined number of successive hops used in routing information coming from the node. For example, if during the characterization time, the node has used the following paths for routing its data to the sink: A-B-C-S – 3 times, A-D-E-F-S – 2 times, A-B-E-F-S – 1 time (A – the node that is sending the data, B, C, … - other nodes in the network, S- sink), we can characterize the routing with the following n-grams (n=3): ABC, BCS, ADE, DEF, EFS, ABE and BEF. In all of the routes, the *n*-gram ABC occurs 3 times, BCS – 3, ADE – 2, DEF - 2, EFS – 3, ABE – 1, BEF – 1. The total number of *n*-grams is 15, so dividing the values given above with 15, we get the frequencies of each *n*-gram which are the values that we assign to out features, i.e. *n*-grams.

It is important to notice that the extracted feature vectors will not be of the same size, so we are not able to use standard distance functions. For this reason, we calculate distance using the approach presented in [5], which calculates distance between sequences.

3.2 Attack Detection

We treat attacks as data outliers and deploy clustering techniques. There are two possible approaches for detecting outliers using clustering techniques [6] depending on the following two possibilities: detecting outlying clusters or detecting outlying data that belong to non-outlying clusters.

The important points necessary for the understanding of the principle is the deployed distance function [5], which is equivalent to Manhattan distance after making the following assumption: the feature that does not exist in the first vector while exists in the second (and vice versa) actually exists with the value equal to 0, since we can say that it occurs with 0 frequency. In this way, we get two vectors of the same size and the distance between the centre and an input is between 0 (when they are formed of the same features with the same feature values) and 2 (when the features with the values greater than 0 are completely different). In the same way, if the set of the features of one is the subset of the feature set of the other, the distance will be between 0 and 1.

In during the testing, different *n*-grams occur in an input, that can happen when the node starts sending data significantly different than before or starts using different routes to send the data, the distance, which is the *QE* value defined previously, between it and its corresponding centre will be greater than 1. This can serve as

evidence of abnormal activities happening in the node or in its routing paths. It is also a typical case when the training is performed with clean data.

3.3 Recovery from Attacks

Every sensor node is being examined by agents that execute clustering algorithms and reside on nodes in its vicinity and listen to its communication. The agents are trained separately. The system of agents is coupled with a reputation system where each node has its reputation value that basically reflects the level of confidence that others have in it based on its previous behavior. In our proposal, the output of an agent affects on the reputation system in the way that it assigns lower reputation to the nodes where it detects abnormal activities and vice versa. We further advocate avoiding any kind of interaction with the low-reputation nodes: to discard any data or request coming from these nodes or to avoid taking them as a routing hop. In this way, compromised nodes remain isolated from the network and have no role in its further performance.

In this work the reputation is calculated in the following way. For the reasons explained in the previous chapter, the value (`rep`) for updating overall reputation based on *QE* is calculated in the following way:

```
1 if (QE<1) rep = 1; 2 else rep=1-QE/2;
```

There are two functions for updating the overall reputation of the node, depending whether the current reputation is below or above the established threshold that distinguishes normal and anomalous behavior. If the current reputation is above the threshold and the node starts behaving suspiciously, its reputation will fall quickly. On the other hand, if the reputation is lower than the established threshold, and the node starts behaving properly, it will need to behave properly for some time until it reaches the threshold in order to "redeem" itself. In order to achieve this, we use the function x+log(1.2*x) because it provides what we want to accomplish: if x is higher than 0.5, the output rises quickly, so the reputation rises; if x is around 0.5, the output is around 0, so the reputation will not change its value significantly; if x is smaller than 0.4, the output falls below 0. Finally, the reputation is updated in the following way:

```
1 if (last_reputation[node]>threshold)
2 new_reputation[node]=last_reputation[node]+rep+log(1.2*rep);
3 else
4 new_reputation[node]=last_reputation[node]+0.05*(rep+log(1.2*rep));
```

If the final value falls out from the [0, 1] range, it is rounded to 0 if it is lower than 0 or to 1 in the opposite case.

However, if during the testing of temporal coherence, we get normal data different from those that the clustering algorithms saw during the training, it is possible to get high *QE* value as well. On the other hand, the spatial coherence should not detect any anomalies. Thus, the final reputation will fall only if both spatial and temporal algorithms detect anomalies. This is implemented in the following way:

```
1 if (value_rep <  threshold)
2 {     if ( space_rep <  threshold) result =    value_rep;
3        else result = 1 - value_rep; }
4 else result = value_rep;
```

where `value_rep` is the reputation assigned by the algorithms for temporal characterization and `space_rep` is the reputation assigned by the algorithms for spatial characterization.

Concerning the detection of routing protocol anomalies, the explained approach can tell us if there is something suspicious in routing paths of a certain node. Yet, it order to find out the nodes that are the origin of the attack, we need to add one more step. In the second step, if the reputation of the routes calculated in the previous step is lower then the established threshold, the hops that participated in bad routes will be added to the global list of bad nodes, or if they already exist, the number of their appearance in bad routes is increased. The similar principle is performed for the correct nodes. For each node, let the number of its appearances in bad routes be *nBad* and the number of its appearances in good routes be *nGood*. Finally, if *nGood* is greater than *nBad*, the node keeps its reputation value, and in the opposite case, it is assigned the following reputation value: *nGood* / (*nGood* + *nBad*). In this way, as the bad node spreads its malicious behavior, its reputation will gradually decrease.

3.4 Developed Techniques

Both SOM [2] and GNG [3] algorithm follow the standard steps. The only problem-specific point is the centre, i.e. node representation and updating. Each centre is implemented as a collection whose size can be changed on the fly and whose elements are the *n*-grams defined in the previous text with assigned occurrence or frequency. The adjustment of nodes (that belong to the map area to be adjusted) is performed in the following way:

- If an *n*-gram of the input instance $v(t)$ exists in the node, its $\varphi(x)$ is modified according to the centre update[2, 3];
- If an *n*-gram of the instance $v(t)$ does not exist in the cluster centre, the *n*-gram is added to the centre with occurrence equal to 1.

4 Experimental Evaluation

The proposed algorithm has been tested on the reputation systems simulator developed by our research group and designed using the C++ programming language. The network consists of a number of distributed nodes. The reputation can be calculated in various ways, which are the implementation of the class *ReputationServer*. The time in the simulator is measured in time ticks, where one tick is equivalent to one sensing period in WSNs. In this work the algorithms have been tested in the presence of the Sybil attack[4], where the compromised node pretends to have multiple IDs, either false, i.e. *fabricated*, or impersonated from other legitimate nodes, i.e. *stolen IDs*. The attacker can affect on many aspects of network (aggregation, routing, etc.).

In the following experiments we will present the performance of the approach in various scenarios, varying the attack strength (Fig. 1) while training with clean data, and varying the starting point of the attack (Fig. 2) while keeping the attack strength. There will be two typical situations: in the first case the attack will start after the end of training, so the training will be performed with "clean" data, while in the second case we will have the situations where the training data contains the traces of attacks as well. The scenario is based on 200 entities that can take one of the possible 2000 positions.

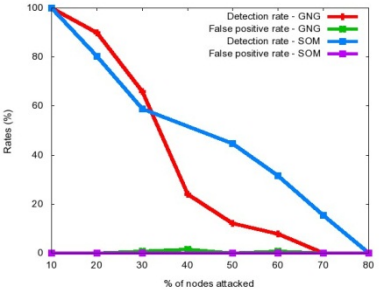


Fig. 1. Detection and false positive rate vs. attack strength

We can observe from Fig. 1 that GNG exhibits higher detection rate than SOM when they deal with the attack of lower strengths (up to 30% of nodes attacked), while for the stronger attacks SOM is clearly the winner. This is probably the result of performing the training with clean data, during which GNG results in having fewer clusters than SOM, thus not being able to detect more subtle differences. On the other hand, when training with a mixture of clean and unclean data, GNG results in having more clusters, thus in this case it is the one being capable of detecting more subtle differences. This is demonstrated in Fig. 2, where we can observe that SOM is capable of isolating the attack if down to 60% of data during the training is normal, while GNG goes down to 5%. Also, the time of isolating the attack of GNG is much lower than the isolation time of SOM. Thus, in this case GNG is clearly the winner.

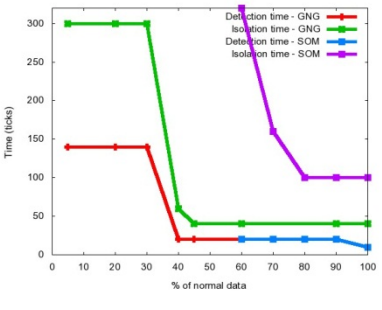


Fig. 2. Detection and isolation times vs. % of normal data during the training

5 Conclusions

In this work we have provided the comparison of SOM and GNG algorithms for clustering-based detection of outliers in its application for security of WSNs. Our results reveal that GNG is superior to SOM when it comes to the level of presence of anomalous data during the training, as GNG is capable of detecting the attack even with small portion of normal data during the training (down to 5%), while SOM need the majority of the training data to be normal (down to 60%) in order to be able to detect it. On the other hand, after both being trained with normal data, SOM performs somewhat better as the attack becomes more aggressive, i.e. it exhibits higher detection rate, although both are capable of detecting the attack in each case.

Thus, based on the presented results, in a general (unknown) case GNG would be more appropriate. However, a combination of both techniques could improve the performances of GNG in certain scenarios. For this reason, in the future we will dedicate our efforts towards obtaining such combination.

Acknowledgments. This work was funded by the Spanish Ministry of Science and Innovation, under Research Grant AMILCAR TEC2009-14595-C02-01.

References

1. Banković, Z., Moya, J.M., Araujo, A., Fraga, D., Vallejo, J.C., de Goyeneche, J.M.: Distributed Intrusion Detection System for WSNs based on a Reputation System coupled with Kernel Self-Organizing Maps. Int. Comp. Aided Design 17(2), 87–102 (2010)
2. Haykin, S.: Neural networks - A comprehensive foundation, 2nd edn. Prentice-Hall (1999)
3. Fritzke, B.: Growing Neural Gas Network Learns Topologies. In: Tesauro, G., Touretzky, D.S., Leen, T.K. (eds.) Advances in Neural Information Processing Systems, vol. 7, pp. 625–632. MIT Press, Cambridge (1995)
4. Roosta, T.G.: Attacks and Defenses on Ubiquitous Sensor Networks, Ph. D. Dissertation. University of California at Berkeley (2008)
5. Rieck, K., Laskov, P.: Linear Time Computation of Similarity for Sequential Data. J. Mach. Learn. Res. 9, 23–48
6. Muñoz, A., Muruzábal, J.: Self-Organizing Maps for Outlier Detection. Neurocomputing 18(1-3), 33–60 (1998)
7. Krontiris, I., Giannetsos, T., Dimitriou, T.: LIDeA: A Distributed Lightweight Intrusion Detection Architecture for Sensor Networks. In: 4th International Conference on Security and Privacy for Communication Networks. ACM (2008)
8. Onat, I., Miri, A.: A Real-Time Node-Based Traffic Anomaly Detection Algorithm for Wireless Sensor Networks. In: Systems Communications, pp. 422–427. IEEE Press (2005)
9. Kaplantzis, S., Shilton, A., Mani, N.: Sekercioglu, Y.A.:Detecting Selective Forwarding Attacks in WSNs using Support Vector Machines. In: Proc. Conf. Int. Sensors, Sensor Networks and Inf., pp. 335–340. IEEE Press (2007)
10. Wallenta, C., Kim, J., Bentley, P.J., Hailes, S.: Detecting Interest Cache Poisoning in Sensor Networks using an Artificial Immune Algorithm. Appl. Intell. 32, 1–26 (2010)
11. Loo, C.E., Ng, M.Y., Leckie, C., Palaniswami, M.: Intrusion Detection for Routing Attacks in Sensor Networks. Int. J. of Dist. Sens. Net. 2(4), 313–332 (2006)

Cryptographic Applications of 3x3 Block Upper Triangular Matrices*

Rafael Álvarez, Francisco Martínez, José-Francisco Vicent, and Antonio Zamora

Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante (Campus de San Vicente)
Ap. Correos 99, E–03080, Alicante, Spain
{ralvarez,fmartine,jvicent,zamora}@dccia.ua.es

Abstract. In this paper we describe a special group of block upper triangular matrices with 3×3 blocks and elements in a finite field. We also verify that, with properly chosen parameters, the cardinality of the subgroup generated by one matrix of this group can be as large as required. Then we introduce two examples of this group of matrices employed in cryptography among the many available: a key exchange scheme and a pseudorandom generator.

Keywords: Block upper triangular matrices key exchange pseudo random generator.

1 Introduction

In this paper, we study and analyze a special group of 3×3 block upper triangular matrices. We also study the order of the subgroup generated since these matrices can generate sets of large and known order with adequately chosen parameters. These matrices have interesting properties for many applications, especially in the field of cryptography (see [5]) since they can be employed to implement several types of primitives.

As an example, we briefly introduce two of them: a key exchange scheme and a pseudorandom generator. Key exchange schemes are useful to setup a private session key for an encrypted communication channel, while pseudorandom generators can be employed to generate keys and other values required in cryptographic protocols and even as stream ciphers (using the Vernam scheme) if they are suitable and fast enough.

2 Description

We describe in this section some basic concepts and notions related to block upper triangular matrices which are necessary for the rest of the paper.

* Partially supported by University of Alicante grants GRE09-02 and GRE10-34 and Generalitat Valenciana grant GV/2011/01.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 97–104, 2012.

2.1 Block Upper Triangular Matrices

Given p a prime number and r, s, t natural numbers, we denote by $\mathrm{Mat}_r(\mathbb{Z}_p)$, $\mathrm{Mat}_s(\mathbb{Z}_p)$, $\mathrm{Mat}_t(\mathbb{Z}_p)$, $\mathrm{Mat}_{r\times s}(\mathbb{Z}_p)$, $\mathrm{Mat}_{r\times t}(\mathbb{Z}_p)$ and $\mathrm{Mat}_{s\times t}(\mathbb{Z}_p)$ the matrices with elements in $\mathbb{Z}_p$, and by $\mathrm{GL}_r(\mathbb{Z}_p)$, $\mathrm{GL}_s(\mathbb{Z}_p)$ and $\mathrm{GL}_t(\mathbb{Z}_p)$ the invertible square matrices also with elements in $\mathbb{Z}_p$.

Let us consider the set Θ of matrices

$$M = \begin{bmatrix} A & Y & X \\ \boldsymbol{0} & B & Z \\ \boldsymbol{0} & \boldsymbol{0} & C \end{bmatrix},$$

where $A \in \mathrm{GL}_r(\mathbb{Z}_p)$, $B \in \mathrm{GL}_s(\mathbb{Z}_p)$, $C \in \mathrm{GL}_t(\mathbb{Z}_p)$, $Y \in \mathrm{Mat}_{r\times s}(\mathbb{Z}_p)$, $X \in \mathrm{Mat}_{r\times t}(\mathbb{Z}_p)$ and $Z \in \mathrm{Mat}_{s\times t}(\mathbb{Z}_p)$.

Theorem 1. *Let*

$$M = \begin{bmatrix} A & Y & X \\ \boldsymbol{0} & B & Z \\ \boldsymbol{0} & \boldsymbol{0} & C \end{bmatrix} \in \Theta,$$

we consider the subgroup generated by the different powers of M. Taking h as a non-negative integer then

$$M^h = \begin{bmatrix} A^h & Y^{(h)} & X^{(h)} \\ \boldsymbol{0} & B^h & Z^{(h)} \\ \boldsymbol{0} & \boldsymbol{0} & C^h \end{bmatrix}, \tag{1}$$

where

$$Y^{(h)} = \begin{cases} \boldsymbol{0}, & \text{if } h = 0 \\ \displaystyle\sum_{i=1}^{h} A^{h-i} Y B^{i-1}, & \text{if } h \geq 1 \end{cases} \tag{2}$$

$$X^{(h)} = \begin{cases} \boldsymbol{0}, & \text{if } h = 0 \\ \displaystyle\sum_{i=1}^{h} (A^{h-i} X C^{i-1} + \sum_{j=1}^{h-i} A^{h-i-j} Y B^{j-1} Z C^{i-1}), & \text{if } h \geq 1 \end{cases} \tag{3}$$

$$Z^{(h)} = \begin{cases} \boldsymbol{0}, & \text{if } h = 0 \\ \displaystyle\sum_{i=1}^{h} B^{h-i} Z C^{i-1}, & \text{if } h \geq 1 \end{cases} \tag{4}$$

Also, if $0 \leq q \leq h$, then

$$Y^{(h)} = A^q Y^{(h-q)} + Y^{(q)} B^{h-q}, \tag{5}$$

$$X^{(h)} = A^q X^{(h-q)} + Y^{(q)} Z^{(h-q)} + X^{(q)} C^{h-q}, \tag{6}$$

$$Z^{(h)} = B^q Z^{(h-q)} + Z^{(q)} C^{h-q}. \tag{7}$$

Proof. The equation is proven using induction on h. For $h = 0$ and $h = 1$, the result is obvious. The equations (2), (3) and (4) are supposed to be true for $h-1$ and will be proven true for h.

$$M^h = MM^{h-1} = \begin{bmatrix} A & Y & X \\ \mathbf{0} & B & Z \\ \mathbf{0} & \mathbf{0} & C \end{bmatrix} \begin{bmatrix} A^{h-1} & Y^{(h-1)} & X^{(h-1)} \\ \mathbf{0} & B^{h-1} & Z^{(h-1)} \\ \mathbf{0} & \mathbf{0} & C^{h-1} \end{bmatrix} = \begin{bmatrix} A^h & Y^{(h)} & X^{(h)} \\ \mathbf{0} & B^h & Z^{(h)} \\ \mathbf{0} & \mathbf{0} & C^h \end{bmatrix}.$$

From the induction hypothesis, applying (2) we have that

$$Y^{(h)} = AY^{(h-1)} + YB^{h-1} = A\sum_{i=1}^{h-1} A^{h-i-1}YB^{i-1} + YB^{h-1}$$
$$= \sum_{i=1}^{h-1} A^{h-i}YB^{i-1} + A^0YB^{h-1} = \sum_{i=1}^{h} A^{h-i}YB^{i-1},$$

in this way, (2) is verified for h.

Finally, in order to demonstrate (3) and (4), we consider that

$$M = \begin{bmatrix} A & Y & X \\ \mathbf{0} & B & Z \\ \mathbf{0} & \mathbf{0} & C \end{bmatrix} = \begin{bmatrix} A_1 & X_1 \\ \mathbf{0} & C_1 \end{bmatrix},$$

where $A_1 = \begin{bmatrix} A & Y \\ \mathbf{0} & B \end{bmatrix}$, $X_1 = \begin{bmatrix} X \\ Z \end{bmatrix}$ and $C_1 = C$.

Attending to the expressions for 2×2 block triangular matrices ([1])

$$M^h = \begin{bmatrix} A^h & X^{(h)} \\ O & B^h \end{bmatrix}, \; X^{(h)} = \begin{cases} \mathbf{0} & if \;\; h = 0 \\ \sum_{i=1}^{h} A^{h-i}XB^{i-1} & if \;\; h \geq 1 \end{cases}$$

we have

$$M^h = MM^{h-1} = \begin{bmatrix} A_1 & X_1 \\ \mathbf{0} & C_1 \end{bmatrix} \begin{bmatrix} A_1^{h-1} & X_1^{(h-1)} \\ \mathbf{0} & C_1^{h-1} \end{bmatrix} = \begin{bmatrix} A_1^h & X_1^{(h)} \\ \mathbf{0} & C_1^h \end{bmatrix},$$

where

$$X_1^{(h)} = \sum_{i=1}^{h} A_1^{h-i}X_1C_1^{i-1} = \sum_{i=1}^{h} \begin{bmatrix} A^{h-i} & Y^{(h-i)} \\ \mathbf{0} & B^{h-i} \end{bmatrix} \begin{bmatrix} X \\ Z \end{bmatrix} C^{i-1}$$
$$= \sum_{i=1}^{h} \begin{bmatrix} A^{h-i}X + Y^{(h-i)}Z \\ B^{h-i}Z \end{bmatrix} C^{i-1}$$
$$= \sum_{i=1}^{h} \begin{bmatrix} A^{h-i}XC^{i-1} + Y^{(h-i)}ZC^{i-1} \\ B^{h-i}ZC^{i-1} \end{bmatrix} = \begin{bmatrix} X^{(h)} \\ Z^{(h)} \end{bmatrix}.$$

We have

$$\begin{aligned} X^{(h)} &= \sum_{i=1}^{h} (A^{h-i} X C^{i-1} + Y^{(h-i)} Z C^{i-1}) \\ &= \sum_{i=1}^{h} (A^{h-i} X C^{i-1} + \sum_{j=1}^{h-i} A^{h-i-j} Y B^{j-1} Z C^{i-1}), \\ Z^{(h)} &= \sum_{i=1}^{h} B^{h-i} Z C^{i-1}, \end{aligned}$$

and the expressions (3) and (4) are proven.

Also, if $0 \leq q \leq h$, we have

$$M^h = M^q M^{h-q} = \begin{bmatrix} A^q & Y^{(q)} & X^{(q)} \\ \mathbf{0} & B^q & Z^{(q)} \\ \mathbf{0} & \mathbf{0} & C^q \end{bmatrix} \begin{bmatrix} A^{h-q} & Y^{(h-q)} & X^{(h-q)} \\ \mathbf{0} & B^{h-q} & Z^{(h-q)} \\ \mathbf{0} & \mathbf{0} & C^{h-q} \end{bmatrix}$$

$$= \begin{bmatrix} A^h & A^q Y^{(h-q)} + Y^{(q)} B^{h-q} & A^q X^{(h-q)} + Y^{(q)} Z^{(h-q)} + X^{(q)} C^{h-q} \\ \mathbf{0} & B^h & B^q Z^{(h-q)} + Z^{(q)} C^{h-q} \\ \mathbf{0} & \mathbf{0} & C^h \end{bmatrix}.$$

Comparing this result to (1) we obtain

$$\begin{aligned} Y^{(h)} &= A^q Y^{(h-q)} + Y^{(q)} B^{h-q}, \\ X^{(h)} &= A^q X^{(h-q)} + Y^{(q)} Z^{(h-q)} + X^{(q)} C^{h-q}, \\ Z^{(h)} &= B^q Z^{(h-q)} + Z^{(q)} C^{h-q}, \end{aligned}$$

that corresponds with the expressions (5), (6) and (7). □

2.2 Set Cardinality

If we construct the blocks A, B and C of a matrix $M \in \Theta$ using primitive polynomials (see [6]), we can guarantee a very high order of the group generated by the powers of M. Let

$$\begin{aligned} f(x) &= a_0 + a_1 x + a_2 x^2 + \ldots + a_{r-1} x^{r-1} + x^r, \\ g(x) &= b_0 + b_1 x + b_2 x^2 + \ldots + b_{s-1} x^{s-1} + x^s, \\ h(x) &= c_0 + c_1 x + c_2 x^2 + \ldots + c_{t-1} x^{t-1} + x^t, \end{aligned}$$

be three primitive polynomials in $\mathbb{Z}_p[x]$, and $\overline{A}, \overline{B}$ y $\overline{C}$ the corresponding companion matrices. Let P, Q and R be three invertible matrices, such that

$$A = P\overline{A}P^{-1}, B = Q\overline{B}Q^{-1} \text{ and } C = R\overline{C}R^{-1}.$$

With this construction we guarantee a maximum order of M

$$o(M) = lcm(p^r - 1, p^s - 1, p^t - 1).$$

Table 1. Order of M as a function of the block sizes and p

p	r	s	t	$o(M)$
3	21	22	23	31
	31	32	33	66
7	14	15	17	38
	30	31	32	77
13	15	17	19	55
	25	26	27	85
19	11	12	13	44
	25	26	27	98
31	9	10	11	42
	25	26	27	114
257	8	9	10	58
	30	31	32	217

Using cyclotomic fields theory we know that the polynomial $x^n - 1$ is divided by $x^d - 1$ if $d|n$; therefore, if we choose r, s and t such that they are relatively prime, the number of common divisors is diminished and the $lcm(p^r - 1, p^s - 1, p^t - 1)$ will be maximal.

As an example, for $p = 13, r = 25, s = 26$ and $t = 27$ we can obtain orders of about 85 decimal digits as seen in table 1, which is adequate for many purposes. Of course, higher values yield better results.

3 Examples

This type of matrices has many uses in the field of cryptography. We briefly introduce here two of them: a key exchange scheme based on asymmetric cryptography and a pseudorandom number generator that can be used to obtain keys, initialization values, etc. or form the basis of a stream cipher using the Vernam construction.

3.1 Key Exchange Scheme

A lot of popular public-key encryption systems are based on number theory problems such as factoring integers or finding discrete logarithms. The underlying algebraic structures are, very often, abelian groups, as we can see in [2,8].

The Discrete Logarithm Problem (DLP, see [3,7,9]) is, together with the Integer Factoring Problem (IFP, see [4]), one of the main problems upon which public-key cryptosystems are built. Thus, efficiently computable groups where the DLP is hard to break are very important in cryptography.

We describe in the following a key exchange scheme using block upper triangular matrices.

$$\text{Let } M_1 = \begin{bmatrix} A_1 & Y_1 & X_1 \\ \boldsymbol{0} & B_1 & Z_1 \\ \boldsymbol{0} & \boldsymbol{0} & C_1 \end{bmatrix} \text{ and } M_2 = \begin{bmatrix} A_2 & Y_2 & X_2 \\ \boldsymbol{0} & B_2 & Z_2 \\ \boldsymbol{0} & \boldsymbol{0} & C_2 \end{bmatrix}$$

be two elements of the set Θ with orders m_1 and m_2 respectively.

If two users U and V wish to exchange a key, they can execute:

1. They agree upon a prime number p and $M_1, M_2 \in \Theta$.
2. U generates two random private keys $e, f \in \mathbb{N}$, computes
$$C = M_1^e M_2^f$$
and sends C to user V.
3. Likewise, V generates two random private keys $v, w \in \mathbb{N}$, computes the matrices
$$F = M_1^v M_2^w$$
and
$$\begin{aligned} D &= M_1^v C M_2^w \\ &= M_1^v M_1^e M_2^f M_2^w \\ &= M_1^{v+e} M_2^{f+w} \\ &= M_1^{e+v} M_2^{w+f} \\ &= M_1^e M_1^v M_2^w M_2^f \\ &= M_1^e F M_2^f \end{aligned}$$
and send D to user U.
4. Finally, user U computes
$$\begin{aligned} (M_1^e)^{-1} D (M_2^f)^{-1} &= (M_1^e)^{-1} M_1^e F M_2^f (M_2^f)^{-1} \\ &= F. \end{aligned}$$

Now both U and V share a common and secret key F.

3.2 Pseudorandom Number Generator

Pseudorandom numbers are very useful in cryptography. They are used as keys, initialization values, etc. Pseudorandom number generators are more attractive than truly random numbers since they are efficiently computable, deterministic, behave in practice as true random numbers and provide easier means for higher statistical quality. For more information on pseudorandom number generators and stream ciphers see [8].

We describe in the following a pseudorandom generator based on the powers of a block upper triangular matrix defined over $\mathbb{Z}_p$, with p prime.

Matrix M is constructed using square blocks in the diagonal of sizes r, s and t respectively,

$$M = \begin{bmatrix} A & Y & X \\ O_{s\times r} & B & Z \\ O_{t\times r} & O_{t\times s} & C \end{bmatrix} . \tag{8}$$

The upper triangular blocks X, Y and Z have sizes $r\times t$, $r\times s$ and $s\times t$ respectively and constitute the initial seed.

Taking the successive powers of matrix M, we obtain a sequence of matrices with a known period as described previously. We process the elements of the upper triangular blocks (X, Y and Z) of each matrix of the sequence to obtain a series of bits with great statistical performance in terms of randomness.

We use expressions (5), (6) and (7) with $k = h - 1$, i.e.

$$X^{(h)} = A^{h-1}X + Y^{(h-1)}Z + X^{(h-1)}C\,, \tag{9}$$
$$Y^{(h)} = A^{h-1}Y + Y^{(h-1)}B\,, \tag{10}$$
$$Z^{(h)} = B^{h-1}Z + Z^{(h-1)}C\,. \tag{11}$$

On each iteration, $M^{(h)}$ is processed by a bit extraction operation to obtain one or more bits per iteration as required. This extraction can work with all the elements of blocks $X^{(h)}$, $Y^{(h)}$ and $Z^{(h)}$.

In order to calculate $X^{(h)}$, $Y^{(h)}$ and $Z^{(h)}$ with (9), (10) and (11) we only need to store A^{h-1}, B^{h-1}, $X^{(h-1)}$, $Y^{(h-1)}$ and $Z^{(h-1)}$, together with the initial B, C, X, Y and Z, because A is used only in the first iteration and the powers of C are never used. This can determine the choice of values for r, s and t.

Moreover, since r, s and t determine the size of seed (the initial values of X, Y and Z) it is important to choose an r, s and t combination such that

$$\log_2(p^{(r\times t)+(r\times s)+(s\times t)}) \geq 128$$

and that

$$\log_2(\mathrm{lcm}(p^r - 1, p^s - 1, p^t - 1)) \geq 128$$

in order to warranty a good enough period for the generator.

4 Conclusions

We have presented a group of block upper triangular matrices with 3×3 blocks and elements in a finite field that have interesting properties for cryptography and described the cardinality of the subgroup generated by one matrix of this group. As example applications we have briefly introduced a key exchange scheme and a pseudorandom number generator with good statistical properties. The specific characteristics of this group of matrices open many interesting possibilities for future work in their application to useful primitives in cryptography: public key cryptosystems, stream ciphers, etc.

References

1. Alvarez, R., Ferrández, F., Vicent, J.F., Zamora, A.: Applying Quick Exponentiation for Block Upper Triangular Matrices. Science Direct 183, 729–737 (2006)
2. Anshel, I., Anshel, M., Goldfeld, D.: An algebraic method for public-key cryptography. Mathematical Research Letters 6, 287–291 (1999)
3. Coppersmith, D., Odlyzko, A., Schroeppel, R.: Discrete logarithms in GF(p). Algorithmica, 1–15 (1986)
4. Diffie, W., Hellman, M.: New directions In Cryptography. IEEE Trans. Information Theory 22, 644–654 (1976)
5. Lee, P.J., Lim, C.H.: Method for Exponentiation in Public-Key Cryptosystems. United States Patent 5,999,627 (1999)
6. Lidl, R., Niederreiter, H.: Introduction to Finite Fields and their Applications. Cambridge University Press (1994)
7. McCurley, K.: The discret logarithm problem. Cryptology and Computational Number Theory. In: Proceedings of Symposia in Applied Mathematics, vol. 42, pp. 49–74 (1990)
8. Menezes, A., Van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Florida (2001)
9. Pohlig, S.C., Hellman, M.E.: An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. IEEE Trans. Info. Theory 24, 106–110 (1978)

Digital Chaotic Noise Using Tent Map without Scaling and Discretization Process

Ruben Vazquez-Medina[1,2], José Luis Del-Río-Correa[3], César Enrique Rojas-López[1], and José Alejandro Díaz-Méndez[4]

[1] Instituto Politécnico Nacional, ESIME-Culhuacan, Santa Ana 1000, 04430, D.F., México
[2] Centro de Física Aplicada y Tecnología Avanzada, UNAM, Boulevard Juriquilla 3001, 76230, Juriquilla, Querétaro, México
[3] Universidad Autónoma Metropolitana Iztapalapa, San Rafael Atlixco 186, 09340, D.F., México
[4] Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro 1, Tonantzintla, Puebla, México
{ruvazquez,crojas}@ipn.mx, jlrc@xanum.uam.mx, ajdiaz@inaoep.mx

Abstract. This work shows how to improve the statistical distribution of signals produced by noise generators designed with one-dimensional chaotic maps. It also shows that in a electronic design the piecewise linear chaotic maps should be considered because they do not have stability islands in its chaotic behavior region, as it occurs with the logistic map commonly used to build noise generators. The design and implementation problems of the noise generators are analyzed and a solution is proposed. This solution relates the tent map output, defined in the real numbers' domain, with a codebook of S elements. The proposed scheme produces digital noise signals using tent map without scaling and discretization process. Finally, this work shows that it is possible to have control over the statistical distribution of the noise signal by selecting the control parameter of the tent map and using, as a design criterion, the bifurcation diagram.

Keywords: Chaotic maps, Noise digital generators, PWLCM.

1 Introduction

The existence of strong pseudo random number generators is highly required in different areas such as cryptography [1][2], steganography [3] and digital watermarking [4]. There are different techniques to generate pseudo random sequences like the architectures based on Linear Feedback Shift Registers (LFSR) [5], Linear Congruential Method (LCM) [6] or Artificial Neural Network (ANN) [7]. Different applications of neural networks to improve pseudo random number generators have been the focus of several research papers because a pseudo random number generator may be created to take advantage of the properties of Multi-Layer Perceptron (MLP) neural networks [8]. Nevertheless, the alternative proposed in this

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 105–115, 2012.

paper uses a method of generating pseudo random sequences based on chaotic functions.

Chaotic signals seem to be very useful in many applications, and especially in the telecommunication area. It has been shown that chaotic sequences can be used in telecommunication systems in order to improve the performance of the system and also increase the security of the communications [9][10][11]. Attractive properties of chaotic signals for this kind of utilization are under study since a long time ago; recently more chaotic signals generators have drawn great attention [12]. Chaotic systems have attractive properties that can be used in pseudo-random noise generation [13][14], such as high sensitivity to initial conditions and mixing property. The 1-D chaotic maps are simple dynamical systems that can be used as iterated functions to implement noise generators easily in hardware and software.

A very useful tool for analyzing the behavior of 1-D chaotic maps is the bifurcation diagram, which can be identified basically by two operating regions: the stable and unstable region. The bifurcation diagram is a tool that allows determining the regions of periodic (stable region) and aperiodic (unstable region) behavior. In some applications it will be necessary to use the region of periodic behavior and in others, like in which they require of pseudorandom sequences, it will be required to use the region of aperiodic (chaotic) behavior. The selection of these regions can be carried out by evaluating the control parameter in the chaotic function. The bifurcation diagram is a formal tool, and in addition it provides a simple and objective view to notice the difference that exists between both types of regions.

The logistic map [15] is a 1-D chaotic map typically used as noise generator when the control parameter is fixed to a maximum value into chaotic region [16]. The logistic map has the disadvantage that can produce periodic signals when it is expected to generate aperiodic signals. Then, the actual behavior of the noise generators based in the logistic map will be very different from its expected behavior. This behavior occurs because it has stability islands within the chaotic region; that is, in the unstable region attractors fixed-point (stability islands) appear according to Charkovsky sequence [17]. Fig 1 shows the Lyapunov exponent and Fig. 2 shows the bifurcation diagram for the logistic map. So, if the control parameter varies significantly, the logistic map can fall into a stability island and then it will generate periodic signals, which is undesirable when designing noise generators. In this way, in a hardware implementation using the logistic map, the control parameter may vary due to manufacturing processes, asymmetry in the transistors of the current mirror circuit, temperature variation, variation of bias voltage, etc. [18].

Notice that when the Lyapunov exponent is the logistic map will produce periodic signals (stable region and stability islands), but when $\lambda \geq 0$ it will produce aperiodic signals (unstable region). When $\lambda \leq 0$ the stability islands appear into the chaotic region, which can be also observed in the bifurcation diagram.

So it is necessary to consider the use of alternatives to overcome these implementation problems. These alternatives can be the piecewise lineal chaotic maps and Tent map is one of them, which have not stability islands and, as a consequence they do not produce periodic signals within the chaotic region. The Fig. 3 shows the bifurcation diagram for tent map. Notice that the bifurcation diagram for tent map has not stability islands.

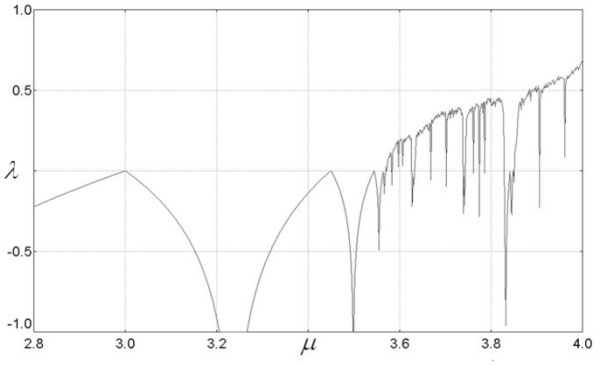


Fig. 1. Lyapunov exponent for the Logistic Map

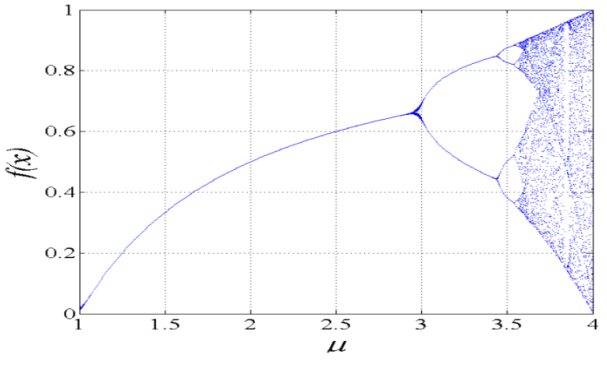


Fig. 2. Bifurcation diagrams for logistic map, μ∈ [1, 4]

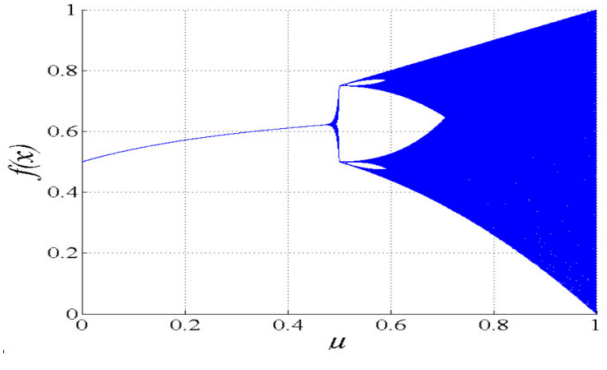


Fig. 3. Bifurcation diagram for tent map, μ∈ [0, 1]

In the other hand, G. Jakimoski et al. [19] suggest using a scaled and discretized logistic map as noise function in a block cryptosystem. Besides the problems that have an implementation for using the logistic map, there are other problems

associated with using a scaled and discretized function. The scaled and discretized function is not a chaotic map; it is only a chaotic map approximation whose quality depends on the precision used in the function and, therefore, it depends of the alphabet size used. Thus, the sequence generated by this procedure may be periodic, although the logistic function is well away from the islands of stability. This condition can be observed in Fig. 4, in which the bifurcation diagram shows low density in the chaotic region when the Logistic Map has been scaled and discretized to the interval $(0, 255) \in \Re$. In similar way, if a piecewise lineal chaotic map (PWLCM) is scaled and discretized its bifurcation diagram will show low density in the chaotic region. Fig. 5 shows the bifurcation diagram for the tent map when it has been scaled and discretized to the interval $(0, 255) \in \Re$. Due to the problems encountered in the use of chaotic maps in noise digital generators for cryptographic systems, in this paper an alternative is proposed which does not require that the chaotic map is scaled and discretized and it uses piecewise lineal chaotic maps, in particular it uses the tent map.

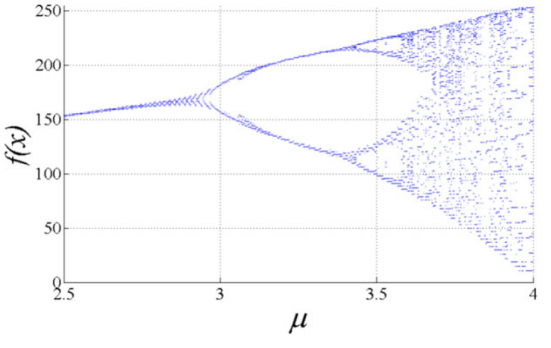


Fig. 4. Bifurcation diagram for the Logistic Map scaled and discretized to $(0, 255) \in \Re$

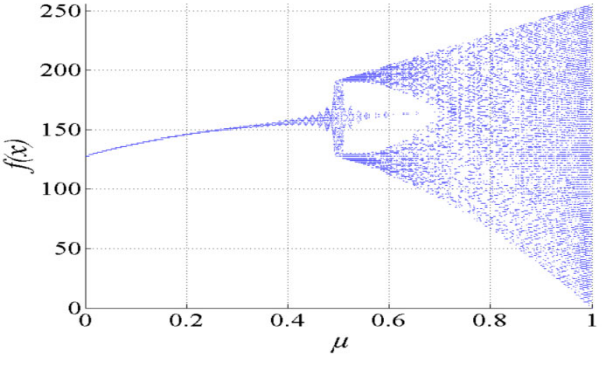


Fig. 5. Bifurcation diagram for the tent map scaled and discretized to $(0, 255) \in \Re$

2 Piecewise Lineal Chaotic Maps: Tent Map

Tent map is a dynamic system that has been extensively studied [20][21] due to its great mathematical simplicity, which allows its implementation in both software and hardware for use in the generation of noise signals [22][23]. These signals can be used as appropriated excitation and training signals in an identifying process [24], in which it is common to use an amplitude modulated pseudo random binary signal (APRBS) or they can be used as an anti-collision mechanism for inventorying processes in Radio Frequency Identification (RFID) systems [25]. Also, these signals can be used in a real option calculation altogether with fuzzy numbers and genetic algorithms for intelligent optimization systems [26]. In this case a tent map has been used to generate pseudorandom numbers, and is determined by the following Eq. 1:

$$x_{n+1} = f_{\mu}(x_n) = \begin{cases} 2\mu x_n + \dfrac{(1 \quad \mu)}{2} & a \le x_n \le \dfrac{1}{2} \\ 2\mu(x_n \quad 1) + \dfrac{(1 \quad \mu)}{2} & \dfrac{1}{2} \le x_n \le b \end{cases} . \tag{1}$$

Which generate the following orbit,

$$\wp_{\Re} = \{x_0, x_1, x_2, \ldots\} \text{ in } [a, b] \in \Re . \tag{2}$$

In this case, a < b ∈ [0, 1] and the control parameter μ ∈ [0, 1] and the initial condition x_0 are chosen arbitrarily but they are known. In order to have a chaotic behavior (unstable region), the tent map require that $0.75 < \mu \le 1$. Fig. 6 shows the statistical distribution of the tent map when $\mu = 1$, which has been calculated by 50 iterations of the tent map over an ensemble of 1000 initial condition arbitrary selected, making use of Birkhoff´s Ergodic Theorem (BET) [27], which implies that it is equivalent to study the evolution of some initial statistical distribution, so that the tent map should be applied to each point in that initial statistical distribution, and when the invariant statistical distribution is obtained, that distribution corresponds to the statistical distribution of the infinite orbit produced by the tent map.

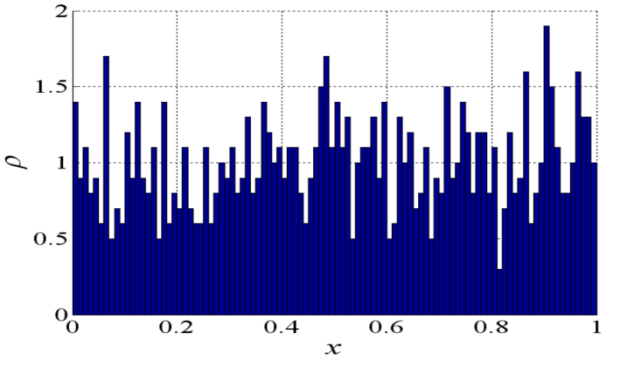


Fig. 6. Statistical distribution of tent map using μ=1

Fig. 7 shows, for each iteration applied to an ensemble of arbitrary selected initial conditions in the tent map, the difference ε between actual and next statistical distribution. Notice that after 10 iterations, ε tends to a constant value. That is, the invariant statistical distribution has been reached.

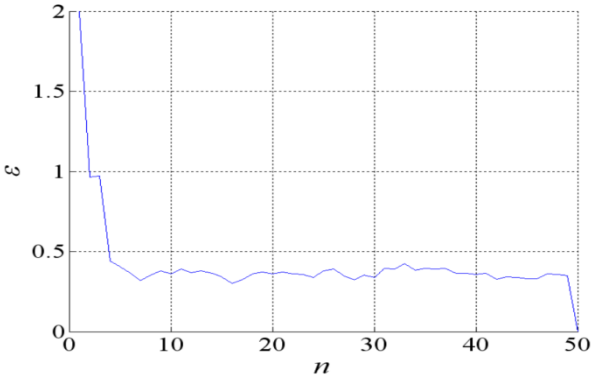


Fig. 7. Difference ε between actual and next statistical distribution in each iteration applied to an ensemble of arbitrary selected initial conditions in the tent map

3 Tent Map Digital Implementation

A digital implementation of this model is possible with certain limitations such as not generating true noise signals because the tent map should be scaled and discretized to the adequate interval according specific application. In this case, the tent map has been scaled and discretized from [0, 1] ∈ ℜ to [0, 255]∈ ℵ, and the new tent map has been defined for an application that will operate with the Extended ASCII alphabet. Therefore after of the scaling and discretization process, the resulting function is not really a chaotic map, but an approximation to it. This phenomenon can be seen in the bifurcation diagram of the tent map (see Fig. 5).

Fig. 8 shows the bifurcation diagram for the tent map when μ ∈ [0.799, 0.801] and the tent map has been scaled to the interval (0, 255) ∈ ℵ. Notice that the sequence produced by this mechanism does not include all the points of the considered interval. Fig. 9 shows the consequence of this situation by the statistical distribution, ρ, which is concentrated only on some values. Notice in Fig. 8 that the map tent only takes 8 values when the value from the parameter μ=0.8. These 8 values in the bifurcation diagram correspond exactly with the 8 values where the statistical distribution is concentrated in Fig. 9, which are limited in an interval approximated of (75, 210). This association condition between the bifurcation diagram and the statistical distribution in the tent map can be maintained for any other values of the parameter μ. This condition demonstrates that the bifurcation diagram is an equivalent tool of analysis to the statistical distribution of the sequence generated with the map tent, with the additional advantage that provides a general view of the statistical behavior for different values from the control parameter μ.

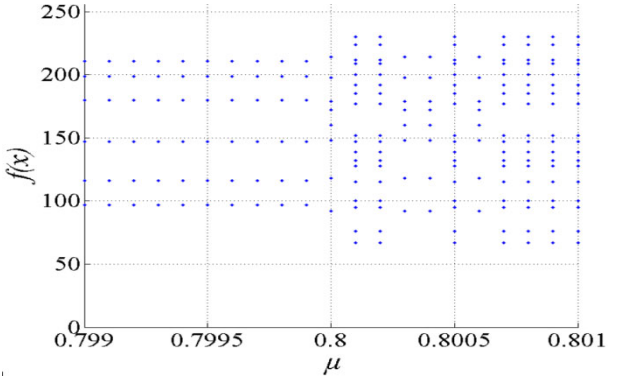


Fig. 8. Bifurcation diagram for scaled and discretized tent map using 8 bits precision and μ ∈ [0.799, 0.801] and c) Statistical distribution when μ=0.8

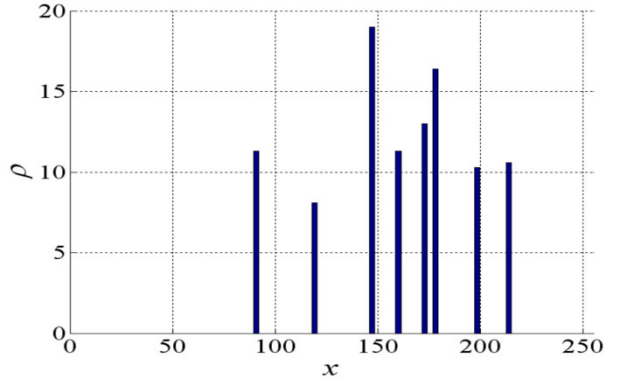


Fig. 9. Statistical distribution for scaled and discretized tent map using 8 bits precision and μ=0.8

4 Map Tent without the Processes of Scaling and Discretization in Digital Applications

Due to the problems encountered in the use of chaotic maps in noise generation, in this work an alternative is proposed in which the chaotic maps do not need to be scaled neither discretized. This alternative consists on defining the chaotic map $f:[a,b]\rightarrow[a,b] \in \Re$ a regular partition of S intervals, in which each interval A_t with t = 0, 1, 2, …, S has a length given by the Eq. 3 where t=0, 1, 2, 3, …255.

$$L\,(A_t) = L = (b\text{-}a)/S\ . \tag{3}$$

In this way, the following orbit is constructed from the chaotic orbit given in the Eq. 2.

$$\wp_{\aleph} = \{y_0, y_1, y_2, \ldots\} \text{ in } [0, 255] \in \aleph \ . \tag{4}$$

$\wp_{\aleph}$ and $\wp_{\Re}$ are related according with the following expression

$$y_m = dec(a_t) \quad if \ \left(I_{At}(x_m)\right) = 1 \ . \tag{5}$$

where m=1, 2, 3, …, t=0, 1, 2, 3, …255, a_t represents the t-th character in the alphabet with 256 elements (Example: Extended ASCII alphabet) and $dec(a_t)$ represents the decimal value of a_t and the membership function $I_{At}(x)$ is given by Eq. 6.

$$I_{At}(x) = \begin{cases} 1, & si \ x \in A_t \\ 0, & si \ x \notin A_t \end{cases} \ . \tag{6}$$

By means of Eq. 4, it is possible to get the bifurcation diagram and observe the behavior of the noise generator proposed. In this case, the aperiodic behavior in the system or unstable region was selected using $\mu > 0.7$ in the bifurcation diagram. Fig. 10 shows for different intervals of μ, the bifurcation diagram for the family of orbits $\wp_{\aleph}$ produced by the mechanism described by Eqs. 4-6 for the tent map defined in $\Re$ and when the scaling and discretization process was not realized. Notice that the behavior of the new orbits (see Fig. 10) is better than the orbits produced by scaled and discretized tent map (see Fig. 8). Fig. 11 shows the statistical distribution for the family of orbits $\wp_{\aleph}$ using μ=0.8 and the results are congruent with the behavior described by the bifurcation diagram of Fig. 10 and they are better than the results of Fig. 9. That is, the statistical distributions have been spread over defining interval. Another way of study the statistical distribution of tent map can be reviewed in [28][29][30].

Comparing the results obtained for the tent map scaled and discretized (See Figs. 8 and 9) with the results obtained with the proposed algorithm (See Figs. 10 and 11), it well-known that the proposed algorithm is a better alternative to generate

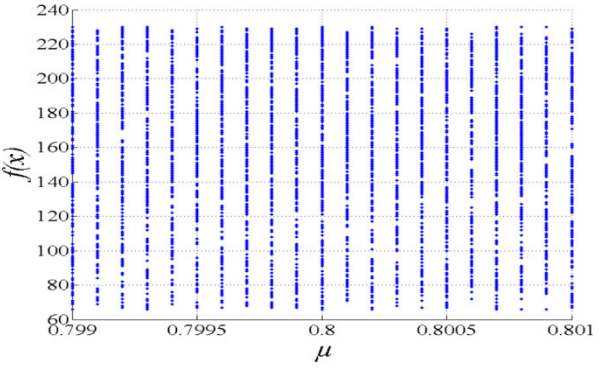


Fig. 10. Bifurcation diagram for the family of orbits $\wp_{\aleph}$ produced by the tent map defined in $\Re$ and the scaling and discretization process was not realized, $\mu \in$ [0.799, 0.801]

pseudorandom sequences. Notice in Fig. 10 that the tent map takes many values ($\gg 8$) when the parameter μ=0.8. These values in the bifurcation diagram correspond exactly with the many values in which the statistical distribution is concentrated in Fig. 11, which are limited in an approximated interval of (65, 230). In this way, the proposed algorithm improves the statistical behavior of the generated sequence, regarding the case in which the tent map is scaled and discretized; the statistical distribution has more components.

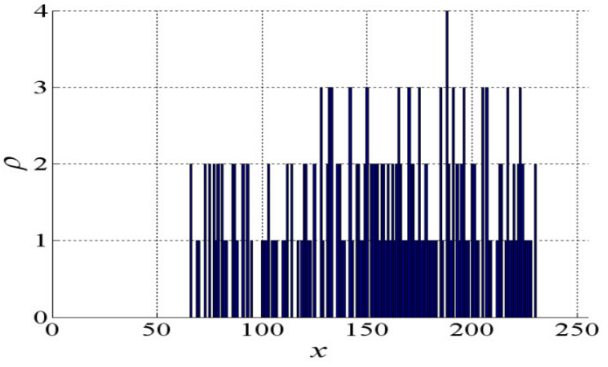


Fig. 11. Statistical distribution using μ=0.8 for the orbits $\wp_{\aleph}$ produced by the tent map defined in $\Re$ when the scaling and discretization process is not realized

5 Conclusions

The discretization and scaling processes do not guarantee that the new map looks a lot like the original map (PWLCM) and therefore it could generate periodic sequences, rather than pseudorandom sequences. The strategies of scaling and discretization of the chaotic maps are very used to adapt functions that generate pseudorandom numbers in the real numbers set at a specific set. Nevertheless, these strategies do not guarantee that the statistical behavior of the function that generates the pseudorandom numbers maintains after the scaling and discretization processes. To reduce this effect, an alternative is to increase the accuracy in the digitization process, but the cost in the electronic implementation could increase significantly. Another approach, which avoids the undesirable effect is the alternative proposed in this paper, this alternative allows discarding the scaling and discretization process, and it only makes an association between a specific alphabet and the map orbit $\wp_{\Re}$ in $\Re$ according to a partition of the PWLCM domain. This procedure generates a new orbit $\wp_{\aleph}$ in $\aleph$, which are digital signals with a statistical distribution closer to an uniform distribution. This condition assures a better statistical behavior the resulting sequence.

Acknowledgments. The authors are grateful to the financial support of the SIP-IPN 20120052 and ICYTDF 270/2010 projects.

References

1. Puczko, M., Yarmolik, V.N.: Designing cryptographic key generators with low power consumption. In: Third IEEE International Workshop on Electronic Design, Test and Applications, DELTA 2006, p. 4 (2006)
2. Zeng, K., Yang, C.-H., Wei, D.-Y., Rao, T.R.N.: Pseudorandom bit generators in stream-cipher cryptography. IEEE Computer 24, 8–17 (1991)
3. Yu, C., Fang, Q., Jianbin, H., Zhong, C.: Density Adjustable Pseudorandom Sequence and its Applications in Information Hiding. In: International Conference on Multimedia Information Networking and Security, MINES 2009, vol. 2, pp. 91–94 (2009)
4. Ahmad, A., Al-Mashari, A., Al-Lawati, A.M.J.: On locking conditions in m-sequence generators for the use in digital watermarking. In: Proceeding of International Conference on Methods and Models in Computer Science, ICM2CS 2009, pp. 1–5 (2009)
5. Arnault, F., Berger, T., Minier, M., Pousse, B.: Revisiting LFSRs for Cryptographic Applications. IEEE Transactions on Information Theory 57, 8095–8113 (2011)
6. Katti, R.S., Kavasseri, R.G.: Secure pseudo-random bit sequence generation using coupled linear congruential generators. In: IEEE International Symposium on Circuits and Systems, SCAS 2008, pp. 2929–2932 (2008)
7. Karras, D.A., Zorkadis, V.: Improving pseudorandom bit sequence generation and evaluation for secure Internet communications using neural network techniques. In: Proceedings of the International Joint Conference on Neural Networks, vol. 2, pp. 1367–1372 (2003)
8. Cheng, L.M., Chan, C.: Pseudorandom generator based on clipped Hopfield neural network. In: Proc. of the IEEE International Symposium on Circuits and Systems, vol. 3, pp. 183–186 (1998)
9. Itoh, M.: Spread spectrum communication via chaos. Int. Journal Bifurcation Chaos 9, 155–213 (1999)
10. Mazzini, G., Setti, G., Rovatti, R.: Chaotic complex spreading sequences for asynchronous DS-CDMA–Part I: System modeling and results. IEEE Trans. Circ. Syst. 44, 937–947 (1997)
11. Kinzel, W., Kanter, I.: Secure Communication with Chaos Synchronization. In: Schoell, E., Schuster, H.G. (eds.) Contribution to Handbook of Chaos Control. Wiley-VCH (2007)
12. Tang, K.W., Tang, W.K.S., Man, K.F.: A chaos-based pseudo-random number generator and its application in voice communications. Int. J. Bifurcat Chaos 17, 923–933 (2007)
13. Rodrigo-Vazquez, A., Espejo-Meana, S.: Analog Building Blocks for Noise and Truly Random Generation in CMOS VLSI. In: Solid-State Circuits Conference, ESSCIRC, Grenoble France, pp. 225–228 (1990)
14. Addabbo, T., Alioto, M., Fort, A., Rocchi, S., Vignoli, V.: Uniform-Distributed Noise Generator Based on a Chaotic Circuit. In: Proceedings of the IEEE Instrumentation and Measurement Technology Conference, IMTC 2006, Sorrento, Italy, pp. 1156–1160 (2006)
15. Peitgen, H.O., Jürgens, H., Dietmar, S.: Chaos and Fractals New Frontiers of Science, USA, p. 864 (2004)
16. Argyris, J., Faust, G., Haase, M.: An Exploration of Chaos: an Introduction for Natural Scientists and Engineers, p. 775. North-Holland, Netherlands (1994)
17. Peitgen, H.O., Jürgens, H., Saupe, D.: Fractals for the Classroom Part II: Complex Systems and Mandelbrot Set, pp. 593–594. Springer, New York (1991)
18. San Martín, J.: Intermittency cascade. Chaos Solitons Fractals 32, 816–831 (2007)
19. Jakimoski, G., Kocarev, L.: Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps. IEEE Transactions on Circuits and Systems I 48, 163–169 (2001)

20. Callegari, S., Setti, G., Langlois, P.J.: A CMOS tailed tent map for the generation of uniformly distributed chaotic sequences. In: IEEE International Symposium on Circuits and Systems, Hong Kong, pp. 781–784 (1997)
21. Nejati, H., Beirami, A., Massoud, Y.: A realizable modified tent map for true random number generation. In: MWSCAS 2008 Conference Proceedings, Knoxville, TN, USA, pp. 621–624 (2008)
22. Addabbo, T., Alioto, M., Bernardi, S., Fort, A., Rocchi, S., Vignoli, V.: The digital Tent map: performance analysis and optimized design as a source of pseudo-random bits. In: IMTC 2004 Conference Proceedings, Como, Italy, pp. 1301–1304 (2004)
23. Huawei, R., Yaz, E.E., Tongyan, Z., Yaz, Y.I.: A generalization of tent map and its use in EKF based chaotic parameter modulation/demodulation. In: 43rd IEEE Conference on Decision and Control, CDC 2004, vol. 2, pp. 2071–2075 (2004)
24. Montiel, O., Castillo, O., Melin, P., Sepúlveda, R.: Evolutionary Optimization of a Wiener Model. In: TI Hybrid Intelligent Systems. Studies in Fuzziness and Soft Computing, pp. 43–58. Springer, Heidelberg (2007)
25. EPC, Radio-Frequency Identity protocols, Class-1 Generation-2 UHF RFID, Protocol for Communicartions at 860 MHz-960MHz, EPC Global Inc., Ver 1.2.0, 46–48 (2008)
26. Pacheco, M.A.C., Vellasco, M.M.B.R.: Intelligent Systems in Oil Field Development under Uncertainty, 139–146 (2009)
27. Lassota, A., Mackey, M.C.: Chaos, Fractals and Noise, p. 64. Springer, New York (1994)
28. Luca, A., Ilyas, A., Vlad, A.: Generating random binary sequences using tent map. In: 10th International Symposium on Signals, Circuits and Systems (ISSCS), pp. 1–4 (2011)
29. Luca, A., Vlad, A., Badea, B., Frunzete, M.: A study on statistical independence in the tent map. In: International Symposium on Signals, Circuits and Systems, ISSCS 2009, pp. 1–4 (2009)
30. Jian-dong, L., Kai, Y., Shu-hong, W.: Coupled Chaotic Tent Map Lattices System with Uniform Distribution. In: 2010 2nd International Conference on e-Business and Information System Security (EBISS), pp. 1–5 (2010)

Hubness-Aware Shared Neighbor Distances for High-Dimensional k-Nearest Neighbor Classification

Nenad Tomašev and Dunja Mladenić

Institute Jožef Stefan
Artificial Intelligence Laboratory
Jamova 39, 1000 Ljubljana, Slovenia
{nenad.tomasev,dunja.mladenic}@ijs.si

Abstract. Learning from high-dimensional data is usually quite a challenging task, as captured by the well known phrase *curse of dimensionality*. Most distance-based methods become impaired due to the distance concentration of many widely used metrics in high-dimensional spaces. One recently proposed approach suggests that using secondary distances based on the number of shared k-nearest neighbors between different points might partly resolve the concentration issue, thereby improving overall performance. Nevertheless, the curse of dimensionality also affects the k-nearest neighbor inference in severely negative ways, one such consequence being known as *hubness*. The impact of hubness on forming shared neighbor distances has not been discussed before and it is what we focus on in this paper. Furthermore, we propose a new method for calculating the secondary distances which is aware of the underlying neighbor occurrence distribution. Our experiments suggest that this new approach achieves consistently superior performance on all considered high-dimensional data sets. An additional benefit is that it essentially requires no extra computations compared to the original methods.

Keywords: Hubness, curse of dimensionality, shared neighbor distances.

1 Introduction

Applied artificial intelligence has become widespread and there is a growing need for reliable intelligent systems. Quite often, hybrid approaches are needed, as problems become more difficult to solve. Learning is one of the basic properties of intelligent systems, so the area of machine learning is cetrainly of prime importance.

Machine learning in many dimensions is often very difficult, due to an interplay of several prohibitive factors. This is usually referred to as the *curse of dimensionality*. In high-dimensional spaces, all data is sparse, as the requirements for proper density estimates rise exponentially with the number of features. Empty space predominates [1] and data lies approximately on the surface of hyper-spheres around cluster means, i.e. in distribution tails. Relative contrast between distances on sample data is known to decrease with increasing dimensionality, i.e. the distances concentrate [2] [3]. The expectation of the distance value increases, but the variance remains constant. It is therefore much more difficult to distinguish between close and distant points. This has a profound impact on nearest neighbor methods, where inference is done based on the k instances

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 116–127, 2012.

most similar (relevant) to the point of interest. The very concept of a nearest neighbor was said to be much less meaningful in high-dimensional data [4].

Difficulty in distinguishing between relevant and irrelevant points is, however, not the only aspect of the dimensionality curse which burdens k-nearest neighbor based inference. The recently described phenomenon of *hubness* has been marked as potentially highly detrimental. The term was coined after *hubs*, very frequent neighbor points which dominate among all the occurrences in the k-neighbor sets of inherently high-dimensional data [5][6]. Most other points either never appear as neighbors or do so very rarely. They are referred to as *anti-hubs*. This property is usually of a geometric nature and does not reflect the semantics of the data, as discussed in [7][8] in the context of music retrieval. The researchers have noticed that some songs are very frequently being retrieved, but were unable to attribute these occurrences to any similarity observable by people.

There is no easy way out, as demonstrated in [9], since dimensionality reduction techniques fail to eliminate the neighbor occurrence distribution skewness for any reasonable dimensionality of the projection space. The skewness decreases only when mapping to very low-dimensional spaces, where too much potentially relevant information is irretrievably lost. Therefore, hubness remains a phenomenon which needs to be taken into account when using nearest neighbor methods on high-dimensional data.

Shared neighbor distances are sometimes used as secondary distance measures when dealing with high-dimensional data, usually in clustering applications [10][11][12][13]. Similarity between points is defined as the number of shared neighbors in their k-neighbor sets, and distances between points are then usually defined in one of the several essentially equivalent ways, which we will address in Section 2.1. Shared neighbor distances have been mentioned as a potential cure for the curse of dimensionality [14].

We have chosen to focus on using the shared neighbor distances in supervised learning, k-nearest neighbor (kNN) classification in particular (where the neighbors are determined based on the secondary distances).

We have measured the hubness of the induced shared neighbor spaces and have shown that hubness-aware k-nearest neighbor classification leads to significant improvements over the basic kNN even when using these secondary distances instead of the original underlying metrics. In other words, shared neighbor distances do not eliminate hubness, so they do not entirely overcome the curse of dimensionality. Hubness has an impact on the forming of the shared neighbor similarity scores, so we propose a new *hubness-aware* method for calculating shared neighbor similarities/distances. This is the main contribution of the paper. Our experiments reveal a consistent and significant improvement when using the newly proposed approach.

The paper is structured as follows. In Section 2 we outline the general motivation for using both the shared neighbor distances and the hubness-aware methods when learning from high-dimensional data, by reviewing some of the recent work in both areas. We proceed by discussing how the two approaches might be successfully combined and propose a new way to define shared neighbor similarities in Section 3. In Section 4 we test our hypothesis on several high-dimensional image datasets and discuss our findings. In the closing section, we summarize all the observations and outline some directions for future research.

2 Related Work

2.1 Shared Neighbor Distances

Regardless of the skepticism expressed in [4], nearest neighbor queries have been shown to be meaningful in high-dimensional data under some natural assumptions [15], at least when it comes to distinguishing between different clusters of data points. If the clusters are pairwise stable, i.e. inter-cluster distances dominate intra-cluster distances, the neighbors will tend to belong to the same cluster as the original point. An obvious issue with this line of reasoning is that cluster assumption violation is present to different degrees in real world data, so that sometimes the categories do not correspond well to the aforementioned clusters. Nevertheless, this observation motivated the researches to consider using secondary distances based on the ranking induced by the original similarity measure [14]. A common approach is to count the number of shared nearest neighbors (SNN) between pairs of points for a given, fixed neighborhood size.

Let $D = (x_1, y_1), (x_2, y_2), ..(x_n, y_n)$ be the data set, where each $x_i \in \mathbb{R}^d$. The x_i are feature vectors which reside in some high-dimensional Euclidean space, and $y_i \in c_1, c_2, ..c_C$ are the labels. Denote by $D_k(x_i)$ the k-neighborhood of x_i. A shared neighbor similarity between two points is then usually defines as:

$$simcos_s(x_i, x_j) = \frac{|D_s(x_i) \cap D_s(x_j)|}{s} \tag{1}$$

where we have used s to denote the neighborhood size, since we will use these similarity measures to perform k-nearest neighbor classification, and the neighborhood sizes in these two cases will be different. The $simcos_s$ similarity can easily be transformed into a distance measure in one of the following ways [14]:

$$\begin{aligned} dinv_s(x_i, x_j) &= 1 - simcos_s(x_i, x_j) \\ dacos_s(x_i, x_j) &= \arccos(simcos_s(x_i, x_j)) \\ dln_s(x_i, x_j) &= -\ln(simcos_s(x_i, x_j)) \end{aligned} \tag{2}$$

All three of the above given distance measures produce the same ranking, so they are equivalent when being used for k-nearest neighbor inference. We based all our subsequent experiments on $dinv_s(x_i, x_j)$.

In shared neighbor distances, all neighbors are treated as being equally relevant. We argue that this view is inherently flawed and that its deficiencies become more pronounced when the dimensionality of the data is increased. Admittedly, there have been some previous experiments on including weights into the SNN framework for clustering [16], but these weights were associated with the positions in the neighbor list, not with neighbor objects themselves. In Section 3 we will discuss the role of hubness in SNN measures.

2.2 Hubs: Very Frequent Nearest Neighbors

High dimensionality gives rise to *hubs*, influential objects which frequently occur as neighbors to other points. Most instances, on the other hand, are very rarely included in

k-neighbor sets, thereby having little or no influence on subsequent classification. What this change in the k-occurrence distribution entails is that potential errors, if present in the hub points, can easily propagate and compromise many k-neighbor sets. Furthermore, hubness is a geometric property of inherently high-dimensional data, as the points closer to the centers of hyper-spheres where most of the data lies tend to become very close to many points and are hence often included as neighbors [6]. This means that hubness of a particular point has little to do with its semantics. Hubs are often not only neighbors to objects of their own category, but also neighbors to many points from other categories as well. In such cases, they exhibit a highly detrimental influence and this is why hubness of the data usually hampers k-nearest neighbor classification.

Hubness-aware algorithms have recently been proposed for clustering [17], instance selection [18], outlier and anomaly detection [9][19] and classification [5][20][21][22], which we will discuss below.

Let us introduce some notation. Denote by $R_k(x_i)$ the reverse neighbor set of x_i, so the number of k-occurrences is then $N_k(x_i) = |R_k(x_i)|$. This total number of neighbor occurrences includes both the *good* occurrences, where the labels of a point and its neighbor match and the *bad* occurrences where there is label mismatch. Formally, $N_k(x_i) = GN_k(x_i) + BN_k(x_i)$, the former being referred to as the good hubness and the latter as the bad hubness of x_i. The bad hubness itself can be viewed as a composite quantity, comprising all the class-specific k-occurrences where label mismatch occurs. Let $N_{k,c}(x_i) = |x \in R_k(x_i) : y = c|$ denote such class-specific hubness. The total occurrence frequency is then simply $N_k(x_i) = \sum_{c \in C} N_{k,c}(x_i)$.

2.3 Hubness-Aware Classification Methods

The basic k-nearest neighbor method [23] is still widely used in many application domains and has been attributed several beneficial asymptotic properties [24][25][26][27]. Hubness in high-dimensional data, nevertheless, affects kNN in some severely negative ways [5][9][6]. This is why several hubness-aware classification algorithms have recently been proposed. An effective vote weighting scheme was first introduced in [5], assigning to each neighbor a weight inversely correlated with its bad hubness. More specifically, $w_k(x_i) = e^{-h_b(x_i,k)}$, where $h_b(x_i, k) = (BN_k(x_i) - \mu_{BN_k})/\sigma_{BN_k}$ is the standardized bad hubness. We will refer to this approach as hubness-weighted k-nearest neighbor (hw-kNN).

Fuzzy measures based on $N_{k,c}(x_i)$ have been introduced in [20], where the fuzzy k-nearest neighbor voting framework was extended to include hubness information (h-FNN). This was further refined in [22] by additionally considering the informativeness of each individual occurrence. Less frequent neighbors were treated as being more informative. The algorithm will be referred to as hubness information k-nearest neighbor (HIKNN). Along with the fuzzy approaches, a naive Bayesian occurrence model was desribed in [21], where the algorithm naive hubness-Bayesian kNN (NHBNN) was proposed for probabilistic k-nearest neighbor classification in high dimensional data. We will see in Section 4.2 that these hubness-aware algorithms are in fact well suited for dealing with secondary SNN distances.

3 Hubness-Aware Shared-Neighbor Distances

Since hubness affects the distribution of neighbors, it must also affect the distribution of neighbors shared between different points. Each x_i is shared between $N_s(x_i)$ data points and participates in $\binom{N_s(x_i)}{2}$ similarity scores. Hub points, by the virtue of being very frequent neighbors, are expected to arise quite frequently as shared neighbors in pairwise object comparisons. What this means, however, is that sharing a hub s-neighbor is quite common and not very informative. This is consistent with observations in [22]. Rarely shared neighbors (anti-hubs), on the other hand, carry information more local to the points of interest and should be given preference when calculating similarities. Figure 1 outlines this observation.

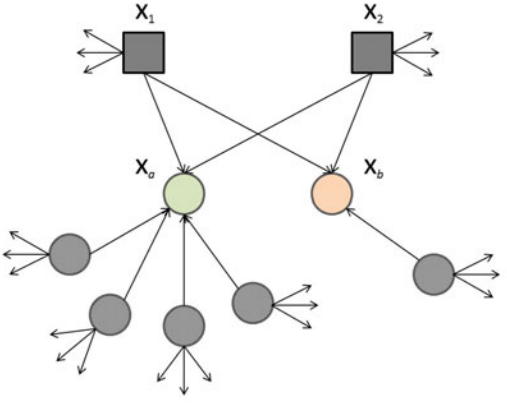


Fig. 1. An illustrative example. x_1 and x_2 share two neighbors, $D_s(x_1) \cup D_s(x_2) = x_a, x_b$. The two shared neighbors are not indicative of the same level of similarity, as x_b is a neighbor only to x_1, x_2 and one other point, while x_a is a more frequently shared neighbor.

One of the most important properties desired in a metric is to allow for good separation between data clusters. This is achieved by minimizing the intra-class distances while maximizing the inter-class distances. Let us compare several types of hub-points from this perspective. There are some hubs which occur almost always as neighbors to points from a single category. Obviously, increasing their weight in the similarity measure would also increase intra-class pairwise similarity. Other hubs occur as neighbors to many different categories inconsistently. Reducing their weight in the similarity measure would certainly reduce inter-class similarity. This is illustrated in Figure 2. The purity of the reverse neighbor sets can clearly be exploited for improving class separation.

In order to refine the basic shared neighbor similarity, we will give preference to less frequent and good neighbor points and reduce the influence of bad hubs. We propose a new SNN similarity measure:

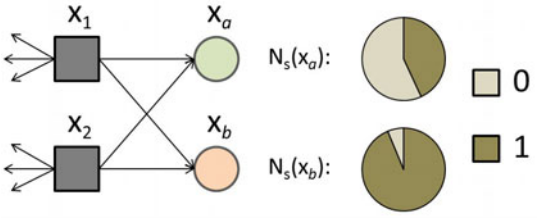


Fig. 2. A binary example where the shared neighbors have significantly different occurrence profiles. x_a is equally often present as a neighbor to objects from both categories, while x_b is almost exclusively in s-neighbor sets of the second class. By favoring x_b over x_a in the similarity score, the average intra-class similarity is expected to increase and the inter-class similarity decreases.

$$simhub_s(x_i, x_j) = \frac{\sum_{x \in D_s(x_i) \cup D_s(x_j)} I_n(x) \cdot (\max H_s - H(R_s(x)))}{s \cdot \max H_s \cdot \max I_n} \tag{3}$$

$$I_n(x) = \log \frac{n}{(N_s(x))}; \qquad \max I_n = \log n \tag{4}$$

$$H(R_s(x)) = H(Y|x \in D_s) = -\sum_{c \in C} \frac{N_{s,c}(x)}{N_s(x)} \log \frac{N_{s,c}(x)}{N_s(x)}; \qquad \max H_s = \log c \tag{5}$$

Though it may seem slightly complicated, $simhub_s$ is in fact very simple and intuitive. The denominator merely serves the purpose of normalization to the $[0, 1]$ range. Each shared neighbor is assigned a weight which is a product of two quantities. Occurrence informativeness ($I_n(x)$) increases the voting weights of rare neighbors. The reverse neighbor set entropy ($H(R_s(x))$) measures the non-homogeneity (inconsistency) in occurrences. When subtracted from the maximum entropy ($\max H_s$), it represents the *information gain* from observing the occurrence of x, under the uniform label assumption. The labels are, of course, not uniformly distributed, but it is convenient to have $(\max H_s - H(R_s(x))) \geq 0$. For the purposes of calculating $I_n(x)$ and $H(R_s(x))$, x is treated as its own 0th nearest neighbor, in order to avoid zero divisions for points which haven't previously occurred as neighbors on the training data. In other words, $N_s(x) := N_s(x) + 1$, $N_{s,y}(x) := N_{s,y}(x) + 1$, where y is the label of x. The $simhub_s$ similarity can be turned into a distance measure in the same way as the $simcos_s$, as previously demonstrated in Eq. 2.

What is great about this new way of defining similarity is that the extra computational cost is negligible, since all the s-neighbor sets need to be calculated anyway. One only has to count the occurrences, which is done in $O(s \cdot n)$ time. Calculating all the $D_s(x)$ neighbor sets accurately takes $\Theta(d \cdot n^2)$ time in high dimensional data, where d is the number of features (since usually $d > s$), which is the time required to compute the distance matrix in the original metric. An approximate algorithm exists, however, which does the same in $\Theta(d \cdot n^{1+t})$, $t \in [0, 1]$ [28]. It is a divide and conquer method based on recursive Lanczos bisection. In our initial experiments, very good estimates are obtained even for $t = 0$ (so, in linear time!), provided that the stop criterion for

subset division is set high enough, since the accurate s-neighborhoods are computed in the leaves of the split.

4 Experiments

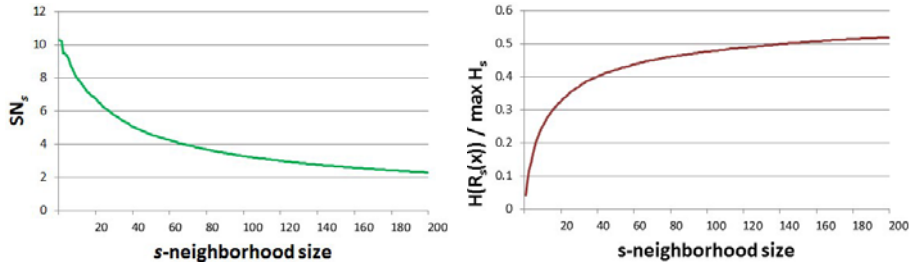
4.1 Overview of the Data

Ten image datasets were selected for the experiments, given that images are both high-dimensional and exhibit significant hubness, which was studied in more detail in [29]. They represent different subsets of the public ImageNet repository (http://www.image-net.org/). More information on the datasets is available in [20]. Images are given as 400-dimensional quantized SIFT feature vectors [30][31] extended by 16-bin color histograms. Both parts of the hybrid representation have been separately normalized. An overview of the datasets is given in Table 1. There is a correspondence between the first and the second five datasets (iNet3..iNet7) and (iNet3Imb..iNet7Imb), as the latter have been obtained from the former via random sub-sampling of the minority classes in order to increase the imbalance in the data. The difference is easily seen in Table 1 by considering the relative imbalance factor: $\mathrm{RImb} = \sqrt{(\sum_{c \in C}(p(c) - 1/C)^2)/((C-1)/C)}$, which is merely the normalized standard deviation of the class probabilities from the absolutely homogenous mean value of $1/c$. Manhattan distance (L_1 metric) was used. The hubness properties are given for $k = 5$ and $k = 50$, since we will be showing the experiments for 5-NN classification on both the primary and 50-SNN secondary distances.

We see that an increase in neighborhood size somewhat reduces the skewness of the occurrence distribution, since more points become hubs. Bad hubness increases, as

Table 1. Summary of datasets. Each dataset is described by the following set of properties: size, number of features, number of classes, for $k = 5$ and $k = 50$: skewness of the k-occurrence distribution (S_{N_k}), the percentage of *bad* k-occurrences (BN_k), the degree of the largest hub-point ($\max N_k$), as well as the relative imbalance of the label distribution and the size of the majority class (expressed as a percentage of the total)

Data set	size	d	C	S_{N_5}	BN_5	$\max N_5$	$S_{N_{50}}$	BN_{50}	$\max N_{50}$	RImb	$p(c_M)$
iNet3	2731	416	3	8.38	21.0%	213	3.10	25.0%	665	0.40	50.2%
iNet4	6054	416	4	7.69	40.3%	204	3.56	46.2%	805	0.14	35.1%
iNet5	6555	416	5	14.72	44.6%	469	6.10	51.1%	1420	0.20	32.4%
iNet6	6010	416	6	8.42	43.4%	275	3.60	51.0%	836	0.26	30.9%
iNet7	10544	416	7	7.65	46.2%	268	4.21	54.3%	1149	0.09	19.2%
iNet3Imb	1681	416	3	3.48	17.2%	75	1.45	21.2%	271	0.72	81.5%
iNet4Imb	3927	416	4	7.39	38.2%	191	3.47	43.2%	750	0.39	54.1%
iNet5Imb	3619	416	5	9.35	41.4%	258	4.61	47.4%	995	0.48	58.7%
iNet6Imb	3442	416	6	4.96	41.3%	122	2.64	48.0%	534	0.46	54%
iNet7Imb	2671	416	7	6.44	42.8%	158	2.72	50.4%	551	0.46	52.1%
AVG	4723.4	416		7.85	37.64%	223.3	3.55	43.8%	797.6	0.36	46.8%



(a) Occurrence skewness (hubness) (b) Reverse neighbor set entropy

Fig. 3. s-occurrence skewness and reverse neighbor set entropy over a range of neighborhood sizes for iNetImb5 dataset

well as the non-homogeneity of reverse neighbors sets. This is illustrated in Figure 3 for iNet5Imb dataset. The degree of major hubs is quite high for $s = 50$ neighborhood size which will be used to generate the secondary SNN distances. In some of the datasets, the major hub appears in approximately 20% of all neighbor lists. This gives justification for the use $I_n(x)$ in $simhub_s$. Likewise, high reverse neighbor set entropies indicate that good hubs are a rarity when using large neighborhoods - so their influence on similarity should be emphasized, whenever possible.

Switching to the secondary distances induces a change in the hubness of the data. We report the hubness-related quantities for both the basic $simcos_s$ similarity and our proposed $simhub_s$ similarity measure. This is summarized in Table 2. A quick comparison with Table 1 reveals that $simcos_s$ does indeed somewhat reduce the bad hubness in the data, but $simhub_s$ reduces it even more, which renders the data easier to handle by k-nearest neighbor algorithms. The average bad $5-$hubness in L_1 is 37.64%. In $simcos_{50}$, 36% of 5-occurrences were label mismatches, while only 33% of occurrences in $simhub_{50}$ were bad. This difference will also be reflected in the classification experiments presented in Section 4.2. Another observation is that both measures reduce the hubness of the data in the original L_1 metric. More hubness remains when using $simhub_{50}$, but the reduction in bad hubness will turn out to be more important.

4.2 Image Classification with Nearest-Neighbor Methods

We have compared $simhub_{50}$ with $simcos_{50}$ in the context of k-nearest neighbor classification on image data. Neighborhood size of 50 was selected after an initial run over several increasing candidate neighborhoods. For $s > 50$, further improvements were minor, so we opted for the lower $s = 50$ instead. We tested the performance of both the basic kNN algorithm and some of the recently proposed hubness-aware algorithms. All tests were done as 10-times 10-fold cross validation and the corrected re-sampled t-test was used to determine statistical significance. Apart from the direct comparison between the two SNN measures, we also conducted experiments using the primary L_1 distance, in order to see how much (if any) improvement is obtained by taking a shared-neighbor approach. The L_1 results are given in Table 3. We see that the hubness-aware

Table 2. Summary of hubness-related quantities as previously given in Table 1, for $simcos_{50}$ and $simhub_{50}$ and $k = 5$. Lower of the two bad hubness values in each line is in bold.

measure:	$simcos_{50}$			$simhub_{50}$		
Data set	S_{N_5}	BN_5	max N_5	S_{N_5}	BN_5	max N_5
iNet3	0.88	19.9%	23	1.34	**17.9%**	30
iNet4	0.83	40.1%	28	1.22	**37.3%**	26
iNet5	1.27	42.9%	38	1.34	**38.6%**	28
iNet6	0.97	41.7%	28	1.29	**38.2%**	31
iNet7	0.86	46.0%	26	1.67	**42.8%**	48
iNet3Imb	0.76	14.6%	19	1.59	**13.3%**	32
iNet4Imb	0.80	37.5%	24	1.15	**34.9%**	31
iNet5Imb	0.94	39.4%	22	1.13	**34.1%**	26
iNet6Imb	0.93	38.3%	25	1.06	**35.1%**	26
iNet7Imb	0.71	40.3%	21	1.09	**37.7%**	26
AVG	0.895	36.07%	25.4	1.288	**32.99%**	30.4

Table 3. Experiments on ImageNet data with the L_1 distance, given for comparison. Classification accuracy is given for kNN, hubness-weighted kNN (hw-kNN), hubness-based fuzzy nearest neighbor (h-FNN), naive hubness-Bayesian kNN (NHBNN) and hubness information k-nearest neighbor (HIKNN). All displayed experiments were performed for $k = 5$. The symbols •/◦ denote statistically significant worse/better performance ($p < 0.05$) compared to kNN. The best result in each line is in bold.

Data set	kNN	hw-kNN	h-FNN	NHBNN	**HIKNN**
iNet3	72.0 ± 2.7	80.8 ± 2.3 ◦	**82.4 ± 2.2** ◦	81.8 ± 2.3 ◦	82.2 ± 2.0 ◦
iNet4	56.2 ± 2.0	63.3 ± 1.9 ◦	**65.2 ± 1.7** ◦	64.6 ± 1.9 ◦	64.7 ± 1.9 ◦
iNet5	46.6 ± 2.0	56.3 ± 1.7 ◦	**61.9 ± 1.7** ◦	61.8 ± 1.9 ◦	60.8 ± 1.9 ◦
iNet6	60.1 ± 2.2	68.1 ± 1.6 ◦	69.3 ± 1.7 ◦	69.4 ± 1.7 ◦	**69.9 ± 1.9** ◦
iNet7	43.4 ± 1.7	55.1 ± 1.5 ◦	**59.2 ± 1.5** ◦	58.2 ± 1.5 ◦	56.9 ± 1.6 ◦
iNet3Imb	72.8 ± 2.4	87.7 ± 1.7 ◦	87.6 ± 1.6 ◦	84.9 ± 1.9 ◦	**88.3 ± 1.6** ◦
iNet4Imb	63.0 ± 1.8	68.8 ± 1.5 ◦	69.9 ± 1.4 ◦	69.4 ± 1.5 ◦	**70.3 ± 1.4** ◦
iNet5Imb	59.7 ± 1.5	63.9 ± 1.8 ◦	64.7 ± 1.8 ◦	63.9 ± 1.8 ◦	**65.5 ± 1.8** ◦
iNet6Imb	62.4 ± 1.7	69.0 ± 1.7 ◦	**70.9 ± 1.8** ◦	68.4 ± 1.8 ◦	70.2 ± 1.8 ◦
iNet7Imb	55.8 ± 2.2	63.4 ± 2.0 ◦	64.1 ± 2.3 ◦	63.1 ± 2.1 ◦	**64.3 ± 2.1** ◦
AVG	59.20	67.64	**69.52**	68.55	69.31

approaches offer significant improvements over the basic kNN method, since these image datasets exhibit high bad hubness.

For further testing on the secondary measures, we will display the results for hw-kNN and HIKNN to represent the hubness-aware algorithms, as one crisp and one fuzzy approach. h-FNN achieves similar results as HIKNN. These comparisons are summarized in Table 4.

It is immediately apparent from the results that the hubness-aware $simhub_{50}$ similarity achieves much better performance than the usually used $simcos_{50}$. The usual SNN approach increased the accuracy of kNN from 59.2% to 63.9%, but the hubness-aware $simhub_{50}$ improved with 68.8%. The difference was statistically significant on

Table 4. Experiments with $simhub_{50}$ and $simcos_{50}$ on ImageNet data. Classification accuracy is given for kNN, hubness-weighted kNN (hw-kNN) and hubness information k-nearest neighbor (HIKNN). All displayed experiments were performed for $k = 5$. The comparisons are done pairwise between the $simhub_{50}$ and $simcos_{50}$ for each classifier, so that the higher value is in bold and •/○ denotes statistically significant worse/better performance of $simhub_{50}$ compared to $simcos_{50}$ ($p < 0.05$).

measure:	$simcos_{50}$			$simhub_{50}$		
Data set	kNN	hw-kNN	HIKNN	kNN	hw-kNN	HIKNN
iNet3	76.9 ± 1.8	81.2 ± 1.8	83.6 ± 1.5	**83.3 ± 1.7** ○	**84.7 ± 1.7** ○	**84.8 ± 1.5**
iNet4	59.2 ± 1.4	63.4 ± 1.4	65.5 ± 1.3	**62.2 ± 1.5** ○	**64.0 ± 4.4**	**65.7 ± 1.4**
iNet5	56.1 ± 1.4	61.8 ± 1.4	64.3 ± 1.3	**63.0 ± 1.2** ○	**66.4 ± 1.3** ○	**67.6 ± 1.3** ○
iNet6	61.2 ± 1.3	68.1 ± 1.3	70.2 ± 1.3	**66.6 ± 1.5** ○	**69.7 ± 1.3**	**70.5 ± 1.3**
iNet7	47.6 ± 1.0	56.6 ± 1.1	59.9 ± 0.9	**56.6 ± 1.1** ○	**60.9 ± 4.3**	**63.0 ± 1.1** ○
iNet3Imb	86.5 ± 1.8	89.2 ± 1.7	89.8 ± 1.6	**88.9 ± 1.6** ○	**89.8 ± 1.6**	**89.9 ± 1.5**
iNet4Imb	67.8 ± 1.6	70.3 ± 1.5	71.2 ± 1.6	**69.7 ± 1.7** ○	**71.2 ± 1.7**	**71.6 ± 1.7**
iNet5Imb	64.8 ± 1.7	67.4 ± 1.5	69.0 ± 1.5	**67.3 ± 1.7** ○	**69.7 ± 1.6** ○	**70.5 ± 1.6**
iNet6Imb	62.3 ± 1.6	69.8 ± 1.5	71.9 ± 1.5	**68.0 ± 1.7** ○	**71.9 ± 1.7**	**73.0 ± 1.8**
iNet7Imb	56.7 ± 1.9	62.7 ± 2.0	65.0 ± 2.2	**62.5 ± 2.0** ○	**65.1 ± 1.9** ○	**65.8 ± 1.9**
AVG	63.91	69.05	71.04	**68.81**	**71.34**	**72.24**

every tested dataset. The advantage of $simhub_{50}$ over $simcos_{50}$ in kNN has the same magnitude as the advantage of $simcos_{50}$ over L_1, approximately 5% increase in accuracy in both cases. Our proposed $simhub_{50}$ similarity does not, however, only improve the classification when used by the basic kNN method. It improves the performance of the tested hubness-aware algorithms as well. kNN with $simhub_{50}$ is actually better than hw-kNN and NHBNN with L_1, but the accuracy of hw-kNN also improves drastically when used with $simhub_{50}$, as shown in Table 4. Noticeable improvement is also present in HIKNN. This shows that taking hubness into account pays off both at the level of primary distances as well as when working with secondary distances. Hubness-aware SNN similarity induces lower bad hubness which does make the classification task somewhat easier, but some bad hubness certainly remains present in the data, and hubness-aware kNN algorithms are well suited for rectifying the remaining detrimental influences. The relative ordering of the three algorithms HIKNN $>$ hw-kNN $>$ kNN seems to be the same in all three experimental settings and is consistent with what was reported in other classification problems [22] [29]. Nevertheless, the total accuracy improvement of both hw-kNN and HIKNN over kNN is reduced from roughly 8.5% and 10% in L_1 to 2.5% and 3.5% in $simhub_{50}$. This is not altogether surprising, since the improvement of HIKNN over kNN has previously been shown to strongly correlate with the bad hubness of the data [22].

5 Conclusions and Future Work

In this paper we proposed a new shared nearest neighbor similarity measure, $simhub_s$, which is aware of the underlying hubs in the primary metric. This is especially important in high-dimensional data, where hubness plays an important role as a nearest neighbor related aspect of the more general curse of dimensionality. As shared neighbor distances have in general been recommended specifically for high-dimensional data,

enriching them with hubness information is even more significant. Our proposed similarity measure, $simhub_s$, assigns a weight to each data object, which aims at minimizing the intra-class distances while maximizing the inter-class distances. This leads to an overall decrease in *bad hubness* of the data, which is highly beneficial.

We test our approach in the context of k-nearest neighbor classification, by comparing the performance of our proposed $simhub_s$ similarity with the standard $simcos_s$ shared neighbor measure. The metrics are compared on high-dimensional, high-hubness image data. Comparisons are made both for the basic kNN as well as the recently proposed hubness aware hw-kNN and HIKNN algorithms. The results clearly demonstrate the advantages of the hubness-aware approach to defining shared nearest neighbor similarity.

The proposed $simhub_s$ approach is not the only way to define hubness-aware weights and we intend to explore other possible directions in our future work. Similarly, we intend to further extend our findings to the unsupervised setting and improve the clustering of high-dimensional data, since this is where the shared neighbor distances are more usually used.

References

1. Scott, D., Thompson, J.: Probability density estimation in higher dimensions. In: Proceedings of the Fifteenth Symposium on the Interface, Amsterdam, pp. 173–179 (1983)
2. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional spaces. In: Proc. 8th Int. Conf. on Database Theory (ICDT), pp. 420–434 (2001)
3. François, D., Wertz, V., Verleysen, M.: The concentration of fractional distances. IEEE Transactions on Knowledge and Data Engineering 19(7), 873–886 (2007)
4. Durrant, R.J., Kabán, A.: When is 'nearest neighbour' meaningful: A converse theorem and implications. Journal of Complexity 25(4), 385–397 (2009)
5. Radovanović, M., Nanopoulos, A., Ivanović, M.: Nearest neighbors in high-dimensional data: The emergence and influence of hubs. In: Proc. 26th Int. Conf. on Machine Learning (ICML), pp. 865–872 (2009)
6. Radovanović, M., Nanopoulos, A., Ivanović, M.: On the existence of obstinate results in vector space models. In: Proc. 33rd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 186–193 (2010)
7. Aucouturier, J.J., Pachet, F.: Improving timbre similarity: How high is the sky? Journal of Negative Results in Speech and Audio Sciences 1 (2004)
8. Aucouturier, J.: Ten experiments on the modelling of polyphonic timbre. Technical report, Docteral dissertation, University of Paris 6 (2006)
9. Radovanović, M., Nanopoulos, A., Ivanović, M.: Hubs in space: Popular nearest neighbors in high-dimensional data. Journal of Machine Learning Research 11, 2487–2531 (2011)
10. Jarvis, R.A., Patrick, E.A.: Clustering using a similarity measure based on shared near neighbors. IEEE Trans. Comput. 22, 1025–1034 (1973)
11. Ertz, L., Steinbach, M., Kumar, V.: Finding topics in collections of documents: A shared nearest neighbor approach. In: Proceedings of Text Mine 2001, First SIAM International Conference on Data Mining (2001)
12. Yin, J., Fan, X., Chen, Y., Ren, J.: High-Dimensional Shared Nearest Neighbor Clustering Algorithm. In: Wang, L., Jin, Y. (eds.) FSKD 2005. LNCS (LNAI), vol. 3614, pp. 494–502. Springer, Heidelberg (2005)

13. Moëllic, P.A., Haugeard, J.E., Pitel, G.: Image clustering based on a shared nearest neighbors approach for tagged collections. In: Proceedings of the 2008 International Conference on Content-Based Image and Video Retrieval, CIVR 2008, pp. 269–278. ACM, New York (2008)
14. Houle, M.E., Kriegel, H.-P., Kröger, P., Schubert, E., Zimek, A.: Can Shared-Neighbor Distances Defeat the Curse of Dimensionality? In: Gertz, M., Ludäscher, B. (eds.) SSDBM 2010. LNCS, vol. 6187, pp. 482–500. Springer, Heidelberg (2010)
15. Bennett, K.P., Fayyad, U., Geiger, D.: Density-based indexing for approximate nearest-neighbor queries. In: ACM SIGKDD Conference Proceedings, pp. 233–243. ACM Press (1999)
16. Ayad, H., Kamel, M.: Finding Natural Clusters using Multi-Clusterer Combiner Based on Shared Nearest Neighbors. In: Windeatt, T., Roli, F. (eds.) MCS 2003. LNCS, vol. 2709, pp. 166–175. Springer, Heidelberg (2003)
17. Tomašev, N., Radovanović, M., Mladenić, D., Ivanović, M.: The Role of Hubness in Clustering High-Dimensional Data. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part I. LNCS, vol. 6634, pp. 183–195. Springer, Heidelberg (2011)
18. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: INSIGHT: Efficient and Effective Instance Selection for Time-Series Classification. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 149–160. Springer, Heidelberg (2011)
19. Tomašev, N., Mladenić, D.: Exploring the hubness-related properties of oceanographic sensor data. In: Proceedings of the SiKDD Conference (2011)
20. Tomašev, N., Radovanović, M., Mladenić, D., Ivanović, M.: Hubness-based fuzzy measures for high dimensional k-nearest neighbor classification. In: Machine Learning and Data Mining in Pattern Recognition Conference, MLDM, New York (2011)
21. Tomašev, N., Radovanović, M., Mladenić, D., Ivanović, M.: A probabilistic approach to nearest neighbor classification: Naive hubness bayesian k-nearest neighbor. In: Proceeding of the CIKM Conference (2011)
22. Tomašev, N., Mladenić, D.: Nearest neighbor voting in high-dimensional data: learning from past occurences. In: PhD forum, ICDM Conference
23. Fix, E., Hodges, J.: Discriminatory analysis, nonparametric discrimination: consistency properties. Technical report, USAF School of Aviation Medicine, Randolph Field, Texas (1951)
24. Stone, C.J.: Consistent nonparametric regression. Annals of Statistics 5, 595–645 (1977)
25. Devroye, L., Györfi, A.K., Lugosi, G.: On the strong universal consistency of nearest neighbor regression function estimates. Annals of Statistics 22, 1371–1385 (1994)
26. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory IT-13(1), 21–27 (1967)
27. Devroye, L.: On the inequality of cover and hart. IEEE Transactions on Pattern Analysis and Machine Intelligence 3, 75–78 (1981)
28. Chen, J., Ren Fang, H., Saad, Y.: Fast approximate kNN graph construction for high dimensional data via recursive Lanczos bisection. Journal of Machine Learning Research 10, 1989–2012 (2009)
29. Tomašev, N., Brehar, R., Mladenić, D., Nedevschi, S.: The influence of hubness on nearest-neighbor methods in object recognition. In: IEEE Conference on Intelligent Computer Communication and Processing (2011)
30. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91 (2004)
31. Zhang, Z., Zhang, R.: Multimedia Data Mining: a Systematic Introduction to Concepts and Theory. Chapman and Hall (2008)

Comparison of Competitive Learning for SOM Used in Classification of Partial Discharge

Rubén Jaramillo-Vacio[1,2], Alberto Ochoa-Zezzatti[3], and Armando Rios-Lira[4]

[1] Comisión Federal de Electricidad-Laboratorio de Pruebas a Equipos y Materiales (LAPEM)
ruben.jaramillo@cfe.gob.mx
[2] Centro de Innovación Aplicada en Tecnologías Competitivas (CIATEC)
[3] Universidad Autónoma de Ciudad Juárez
[4] Instituto Tecnológico de Celaya

Abstract. This paper shows different competitive learning algorithms for Self Organizing Map (SOM) and are experimentally compared, the characterization of the obtainable results in terms of quality of SOM. The competitive learning algorithms showed to SOM algorithm are Winner-takes-all, Frequency Sensitive Competitive Learning and Rival Penalized Competitive Learning. As a case study: the performance in classification of partial discharge on power cables.

Keywords: Competitive learning, Self Organizing Maps, Partial Discharge, Quality Measurements, Diagnosis.

1 Introduction

Competitive learning is an efficient tool for Self Organizing Maps, widely applied in variety of signal processing problems such as classification, data compression, etc.

In the field of data analysis two terms frequently encountered are supervised and unsupervised clustering methodologies. While supervised methods mostly deal with training classifiers for known symptoms, unsupervised clustering provides exploratory techniques for finding hidden patterns in data. With the huge volumes of data being generated from the different systems everyday, what makes a system intelligent is its ability to analyze the data for efficient decision-making based on known or new cluster discovery. The partial discharge (PD) is a common phenomenon which occurs in insulation of high voltage, this definition is given in [1]. In general, the partial discharges are in consequence of local stress in the insulation or on the surface of the insulation.

The typical competitive learning algorithm *k*-means (or called hard *c*-means) clustering is a batch algorithm for designing a vector quantizer, which is a mapping of input vectors to one of *c* predetermined codevectors (also called codebooks) [2]. Fuzzy *c*-means (FCM) clustering is a fuzzy extension of hard *c*-means clustering. The FCM and its varieties have been widely studied and applied in various areas [3-5]. During the last fifteen years there have been developed new advanced algorithms that eliminate the ”dead units” problem and perform clustering without predicting the

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 128–138, 2012.

exact cluster number, as for example: the frequency competitive algorithm (FSCL) [6], the incremental *k*-means algorithm, the rival penalizing competitive algorithm (RPCL) [7].

We evaluate the performance of algorithms in which competitive learning is applied of partial discharge dataset, quantization error, topological error and time in seconds per training epoch. The result from classification of PD shows that *Winner-takes-all* (WTA) has better performance than Frequency Sensitive Competitive Learning (FSCL) and Rival Penalized Competitive Learning (RPCL).

Table 1 show a concentration of researchers who worked on the feature extraction, recognition and classification of PD, as well as the different artificial intelligent tools and the constraint to utilize these methods.

Table 1. Classification and diagnosis in PD using Data Mining Tools

Authors	Tool and Objective	Constraints
Mozroua *et al* [8] Kravida [9]	Tool: Supervised Neural Networks. Objective: Recognition between different sources formed of cylindrical cavities.	Recognition of different sources in the same sample
Kim *et al* [10]	Tool: Fuzzy-Neural Networks. Objective: Comparison between Back Propagation Neural Network and Fuzzy-Neural Networks	Performance in the case of multiple discharges and including defects and noises.
Ri-Cheng *et al* [11]	Tool: Particle Swarm Optimization Objective: Localization of PD in the power transformer.	On site application should improve performance.
Chang *et al* [12]	Tool: Self Organizing Map (SOM). Objective: PD pattern recognition and classification.	Quality and optimization structure of SOM.
Fadilah Ab Aziz *et al*[13]	Tool: Support Vector Machine (SVM). Objective: Feature Selection and PD classification.	SVM is not reliable for small dataset.
Hirose *et al* [14]	Tool: Decision Tree Objective: Feature Extraction and PD classification	The allocation rules are sensitive to small perturbations in the dataset (Instability)

This discovered knowledge then forms the basis of two separate decision support systems for the condition assessment/defect classification of these respective plant items. In this paper is shown a comparative of competitive learning algorithms to classify measured PD activities into underlaying insulation defects or source that generate PD's using Self Organizing Maps (SOM). Multidimensional scaling (MDS) is a nonlinear feature extraction technique [15], it aims to represent a multidimensional dataset in two or three dimensions such that the distance matrix in the original *k*-dimensional feature space is preserved as faithfully as possible in the projected space. The SOM, or Kohonen Map [16], can also be used for nonlinear feature extraction. It should be emphasized that the goal here is not to find an optimal

clustering for the data but to get good insight into the cluster structure of the data for data mining purposes. Therefore, the clustering method must be fast, robust, and visually efficient.

2 Partial Discharge: Concepts

Partial discharges occur wherever the electrical field is higher than the breakdown field of an insulating medium: Air: 27 kV/cm (1 bar), SF6: 360 kV/cm (4 bar), Polymers: 4000 kV/cm.

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

- **Corona Discharges** – Occurs in gases or liquids caused by concentrated electric fields at any sharp points on the electrodes.
- **Internal Discharges** – Occurs inside a cavity that is surrounded completely by insulation material; might be in the form of voids (e.g. dried out regions in oil impregnated paper-cables).
- **Surface Discharges** – Occurs on the surface of an electrical insulation where the tangential field is high e.g. end windings of stator windings.

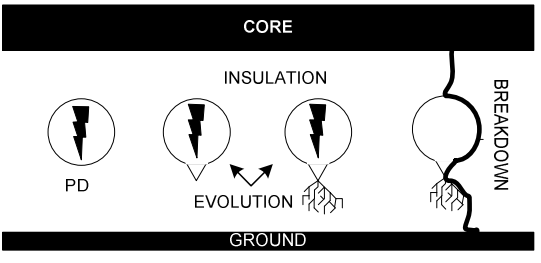


Fig. 1. Example of damage in polymeric power cable from the PD in a cavity to breakdown

In general, the partial discharges are in consequence of local stress in the insulation or on the surface of the insulation. This phenomenon has a damaging effect on the equipments, for example transformers, power cables, switchgears, and others (see Figure 1). The first approach in a diagnosis is selecting the different features to classify measured PD activities into underlying insulation defects or source that generate PD's. The partial discharge measurement is a typical nondestructive test and it can be used to judge the insulation performance at the beginning of the service time taking into account the reduction of the performance during the service time by ageing, whereby the ageing depends on numerous parameters like electrical stress, thermal stress and mechanical stress. In particular for solid insulation like XLPE on power cables where a complete breakdown seriously damages the test object the partial discharge measurement is a tool for quality assessment. The charge that a PD generates in a cavity is called the physical charge and the portion of the cavity surface

that the PD affects is called the discharge area. $E_{applied}$ is the applied electric field and $q_{physical}$ is the physical charge [17].

The pulse repetition rate n is given by the number of partial discharge pulses recorded in a selected time interval and the duration of this time interval. The recorded pulses should be above a certain limit, depending on the measuring system as well as on the noise level during the measurement. The pulse repetition frequency N is the number of partial discharge pulses per second in the case of equidistant pulses. Furthermore, the phase angle ϕ and the time of occurrence t_i are information on the partial discharge pulse in relation to the phase angle or time of the applied voltage with period T. For PD diagnosis test, is very important to classify measured PD activities, since PD is a stochastic process, namely, the occurrence of PD depends on many factors, such as temperature, pressure, applied voltage and test duration; moreover PD signals contain noise and interference [18]. Therefore, the test engineer is responsible for choosing proper methods to diagnosis for the given problem. In order to choose the features, it is important to know the different source of PD, an alternative is though pattern recognition. This task can be challenging, nevertheless, features selection has been widely used in other field, such as data mining [19] and pattern recognition using neural networks [8-10]. This research only shows test on laboratory without environment noise source, and it is a condition that does not represent the conditions on site, Markalous [20] presented the noise levels on site based on previous experiences.

The phase resolved analysis investigates the PD pattern in relation to the variable frequency AC cycle. The voltage phase angle is divided into small equal windows. The analysis aims to calculate the integrated parameters for each phase window and to plot them against the phase position (ϕ).

- $(q_m - \phi)$: the peak discharge magnitude for each phase window plotted against ϕ, where q_m is peak discharge magnitude.

3 Self Organizing Map (SOM)

3.1 Winner Takes All

The Self Organizing Map developed by Kohonen, is the most popular neural network models. The SOM algorithm [15,21] is based on unsupervised competitive learning called winner – takes – all, which means that the training in entirely data-driven and that the neurons of the map compete with each other.

Supervised algorithms [8, 9] like multi-layered perceptron, required that the target values for each data vector are known, but the SOM does not have this limitation. The SOM is a neural network model that implements a characteristics non-linear mapping from the high-dimensional space of input signal onto a typically 2-dimensional grid of neurons. The SOM is a two-layer neural network that consists of an input layer in a line and an output layer constructed of neurons in a two-dimensional grid.

The neighborhood relation of neuron i, an n-dimensional weight vector w is associated; n is the dimension of input vector. At each training step, an input vector **x**

is randomly selected and the Euclidean distances between **x** and **w** are computed. The image of the input vector and the SOM grid is thus defined as the nearest unit w_{ik} and best-matching unit (BMU) whose weight vector is closest to the **x** [21]:

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

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

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

where **c** denote the neighborhood kernel around the best-matching unit and α is the learning rated and δ is the neighborhood function.

The number of panels in the SOM is according the A x B neurons, the U-matrix representation is a matrix U ((2A-1) x (2B-1)) dimensional [22]. The selection of the distance criterion depends on application. In this paper, Euclidean distance is used because it is widely worn with SOM [23].

It is complicated to measure the quality of an SOM. Resolution and topology preservation are generally used to measure SOM quality [24]. There are many ways to measure them. The quantization error (qe) is calculated to measure the quality of the map. The quantization error qe is the average distance between each data vector and its BMU, measuring map resolution. The topological error te is the proportion of all data vectors for which first and second BMUs are adjacent units, otherwise this is regarded as violation of topology and thus penalized by increasing the error value. For recognition and classification of partial discharge patterns is important to minimize errors and processing time, however, the parameter that will succeed in the diagnosis is the quantization error because it evaluated the codebook.

3.2 The Frequency Sensitive Competitive Algorithm

The k-means algorithm has also the "dead units" problem, which means that if a centre is inappropriately chosen, it may never be updated, thus it may never represent a class.

To solve the "dead units" problem it has been introduced the so called "frequency sensitive competitive learning" algorithm (FSCL) [25] or competitive algorithm "with conscience". Each centre counts the number of times when it has won the competition and reduces its learning rate consequently. If a center has won too often "it feels guilty" and it pulls itself out of the competition. The FSCL algorithm is an extension of *k*-means algorithm, obtained by modifying relation (2) according to the following one:

$$j = \arg\min \gamma_i \left\| x(n) - c_i(n) \right\| \qquad i = 1, ..., N \tag{4}$$

where n is the inputs, N represents the centres numbers, the relative winning frequency γ_i of the centre c_i defined as:

$$\gamma_i = \frac{s_i}{\sum_{i=1}^{n} s_i} \tag{5}$$

where s_i is the number of times when the centre c_i was declared winner in the past. So the centers that have won the competition during the past have a reduced chance to win again, proportional with their frequency term γ. After selecting out the winner, the FSCL algorithm updates the winner with next equation:

$$c_i(n+1) = c_i(n) - \eta\left[x(n) - c_i(n)\right] \tag{6}$$

η is the learning rate, in the same way as the *k*-means algorithm, and meanwhile adjusting the corresponding s_i with the following relation:

$$s_i(n+1) = s_i(n) + 1 \tag{7}$$

3.3 The Rival Penalized Competitive Learning Algorithm

The rival penalized competitive algorithm (RPCL) [25] performs appropriate clustering without knowing the clusters number. It determines not only the winning centre *j* but also the second winning center *r*, named rival

$$r = \arg\min \gamma_i \left\| x(n) - c_i(n) \right\|, \quad i = 1, ..., N \;\; i \neq j \tag{8}$$

The second winning centre will move away its centre from the input with a ratio β, called the de-learning rate. All the other centres vectors will not change. So the learning law can be synthesized in the following relation:

$$c_i(n+1) = \begin{cases} c_i(n) + \eta\left[x(n) - c_i(n)\right] & if \;\; i = j \\ c_i(n) - \beta\left[x(n) - c_i(n)\right] & if \;\; i = j \\ c_i(n) & if \;\; i \neq j \;\; and \;\; i \neq r \end{cases} \tag{9}$$

If the learning speed η is chosen much greater than β, with at least one order of magnitude, the number of the output data classes will be automatically found. In other words, suppose that the number of classes is unknown and the centres number *N* is greater than the clusters number, than the centres vectors will converge towards the centroids of the input data classes. The RPCL will move away the rival, in each iteration, converging much faster than the *k*-means and the FSCL algorithms.

4 Analysis of PD Data

PD measurements for power cables are generated and recorded through laboratory tests. Corona was produced with a point to hemisphere configuration: needle at high voltage and hemispherical cup at ground. Surface discharge XLPE cable with no stress relief termination applied to the two ends. High voltage was applied to the cable inner conductor and the cable sheath was grounded, this produces discharges along the outer insulation surface at the cable ends. Internal discharge was used a power cable with a fault due to electrical treeing. Were considered the pattern characteristic of univariate phase-resolved distributions as inputs, the magnitude of PD is the most important input as it shows the level of danger, for this reason the input in the SOM the raw data is the peak discharge magnitude for each phase window plotted against ($qm - \phi$). Figure 2 shows the conceptual diagram training. In the cases analyzed, the original dataset is 1 million of items, was used a neurons array of 10×10 cells to extract features. As it is well known, in fact, a too small number of neurons per class could be not sufficient to represent the variability of the samples to be classified, while a too large number in general makes the net too much specialized on the samples belonging to the training set and consequently reduces its generalization capability. Moreover a too large number of neuron per class implies a long training time and a possible underutilization of some of the neural units.

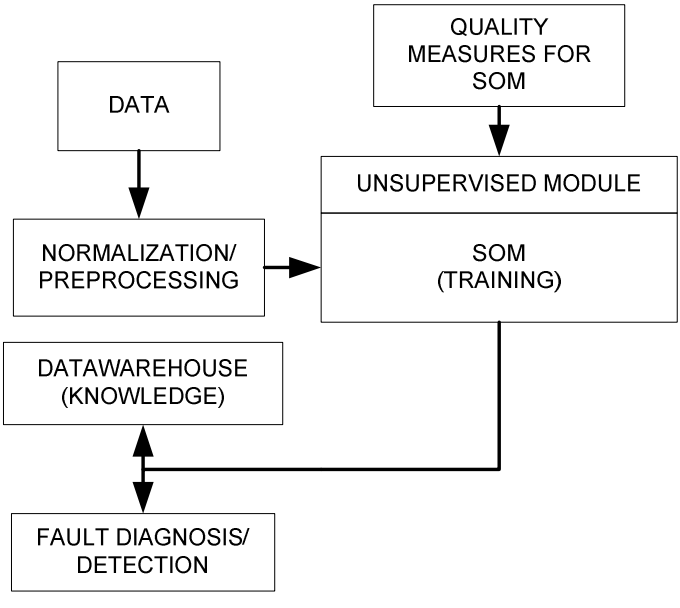


Fig. 2. The component interaction between SOM

In table 2 are shower the parameters for training to each competitive learning algorithm. The coefficient γ has been dynamically changed during the training.

Table 2. Parameters for training

	WTA	FSCL	RPCL
Epoch	100	100	100
η	0.1	0.1	0.05
β	0.01	0.01	0.01

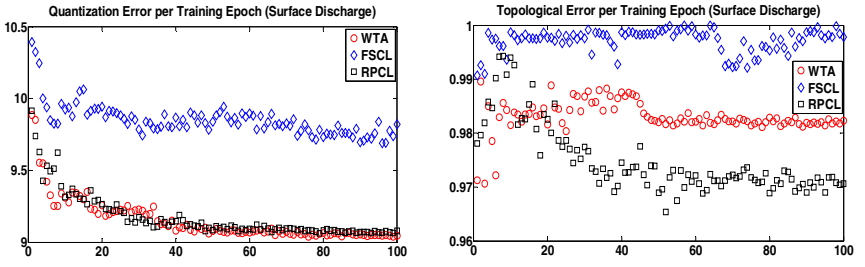


Fig. 3. Quantization and Topological error per Training Epoch (Surface Discharge)

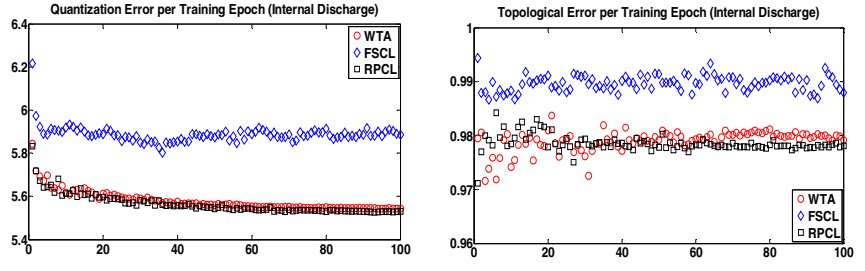


Fig. 4. Quantization and Topological error per Training Epoch (Internal Discharge)

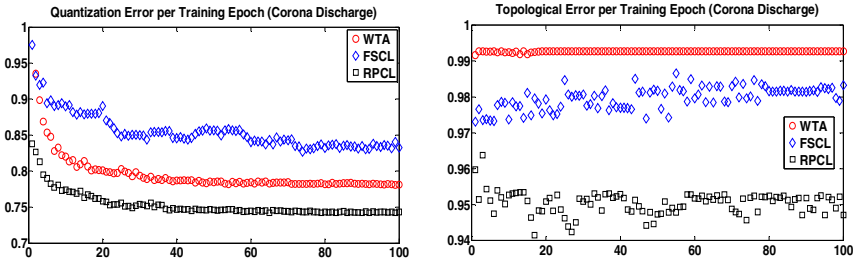


Fig. 5. Quantization and Topological error per Training Epoch (Corona Discharge)

In figure 3, 4 and 5 is showed the performance of the competitive learning algorithms to different PD source.

We evaluate the performance of algorithms in which competitive learning is applied of partial discharge dataset, quantization error, topological error and time in seconds per training epoch are showed in table 3, and it is observer that the WTA works with less time of training (figure 6), but the FSCL is not always satisfactory because the training time is long and more topological error (figure 6 and 7). The RCPL have longest training time but is the algorithm with less topological error. The comparison was evaluated using Tukey–Kramer method [26], is a single-step multiple comparison procedure and statistical test generally used to find which means are significantly different from one another.

Table 3. Performance of Training in Som

	WTA	**FSCL**	**RPCL**
Surface Discharge			
$\mathbf{q_e}$	9.0	9.8	9.1
$\mathbf{t_e}$	0.985	1	0.97
time	849 seconds	1160 seconds	1226 seconds
Internal Discharge			
$\mathbf{q_e}$	5.5	5.9	5.4
$\mathbf{t_e}$	0.98	0.99	0.98
time	173 seconds	222 seconds	241 seconds
Corona Discharge			
$\mathbf{q_e}$	0.78	0.85	0.75
$\mathbf{t_e}$	0.99	0.98	0.95
time	889 seconds	1191 seconds	1362 seconds

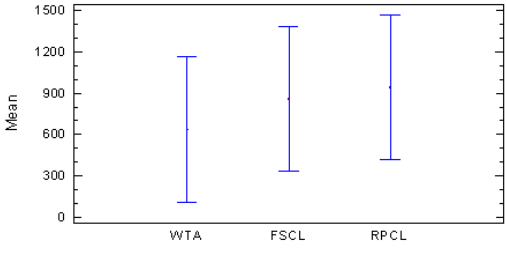


Fig. 6. Comparison the training time

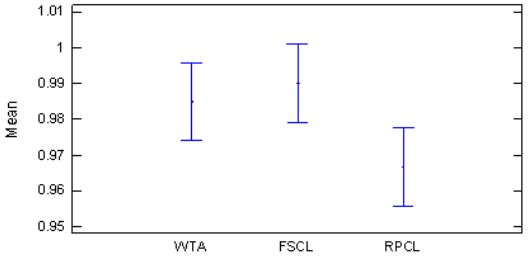


Fig. 7. Comparison of the Topological error (te)

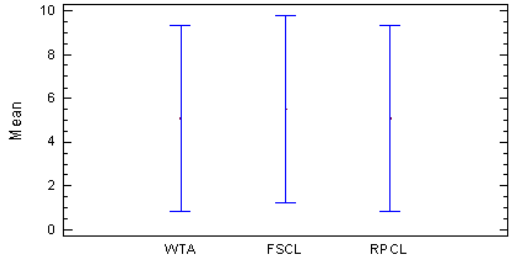


Fig. 8. Comparison of the Quantization error (qe)

5 Conclusion

PD patterns recognition and classification require an understanding of the traits commonly associated with the different source and relationship between observed PD activity and responsible defect sources. This paper shows the performance of SOM using different competitive learning algorithms to classify measured PD activities into underlaying insulation defects or source that generate PD's, its showed that RPCL is the better algorithm with less topological error, but its overall performance are not always satisfactory, being alternative in accord at the performance FSCL or WTA algorithms.

References

1. IEC 60270 Ed. 2. High-voltage test techniques - Partial discharge measurements, 15–16 (2000)
2. Pollard, D.: Quantization and the method of k-means. IEEE Transaction on Information Theory, 28–199 (1982)
3. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press (1981)
4. Yang, M.S.: A survey of fuzzy clustering. Math. Comput. Model (1993)
5. Höppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis: Methods for Classification Data Analysis and Image Recognition. Wiley, New York (1999)

6. Wang, X., Liu, H., Lu, J., Yohagy, T.: Combining recurrent neural networks with self-organizing maps for channel equalization. IEEE Trans. on Communications E85-E, 2227–2235 (2002)
7. Xu, L., Krzyzak, A., Oja, A.E.: Rival penalized competitive learning for clustering analysis. rbf net and curve detection. IEEE Trans. on Neural Networks (4), 636–664 (1993)
8. Mazroua, A.: PD pattern recognition with neural netwoks using the multilayer perception technique. IEEE Transactions on Electrical Insulation 28, 1082–1089 (1993)
9. Krivda, A.: Automated Recognition of Partial Discharge. IEEE Transactions on Dielectrics and Electrical Insulation 28, 796–821 (1995)
10. Kim, J., Choi, W., Oh, S., Park, K., Grzybowski, S.: Partial Discharge Pattern Recognition Using Fuzzy-Neural Networks (FNNs) Algorithm. 272–275 (2008)
11. Ri-Cheng, L., Kai, B., Chun, D., Shao-Yu, L., Gou-Zheng, X.: Study on Partial Discharge Localization by Ultrasonic Measuring in Power Transformer Based on Particle Swarm Optimization. In: International Conference on High Voltage Engineering and Application, pp. 600–603 (2008)
12. Chang, W., Yang, H.: Application of Self Organizing Map Approach to Partial Discharge Pattern Recognition of Cast-Resin Current Transformers. WSEAS Transaction on Computer Research 3(3), 142–151 (2008)
13. Ab Aziz, N.F., Hao, L., Lewin, P.: Analysis of Partial Discharge Measurement Data Using a Support Vector Machine. In: 5th Student Conference on Research and Development, pp. 1–6 (2007)
14. Hirose, H., Hikita, M., Ohtsuka, S., Tsuru, S., Ichimaru, J.: Diagnosis of Electric Power Apparatus using the Decision Tree Method. IEEE Transactions on Dielectrics and Electrical Insulation 15, 1252–1260 (2008)
15. Kantardzic, M.: Data Clustering, Theory, Algorithms and Methods, pp. 53–57. ASA-SIAM (2007)
16. Vesanto, J.: Data Exploration Process Based on the Self Organizing Map. Doctoral Thesis in Computer Science. Helsinki University of Technology (2002)
17. Forssén, C.: Modelling of cavity partial discharges at variable applied frequency. Sweden: Doctoral Thesis in Electrical Systems. KTH Electrical Engineering (2008)
18. Edin, H.: Partial discharge studies with variable frequency of the applied voltage. Sweden: Doctoral Thesis in Electrical Systems. KTH Electrical Engineering (2001)
19. Lai, K., Phung, B.: Descriptive Data Mining of Partial Discharge using Decision Tree with genetic algorithms. AUPEC (2008)
20. Markalous, S.: Detection and location of Partial Discharges in Power Transformers using acoustic and electromagnetic signals. Stuttgart University: PhD Thesis (2006)
21. Kohonen, T.: Engineering Applications of Self Organizing Map. Proceedings of the IEEE (1996)
22. Rubio-Sánchez, M.: Nuevos Métodos para Análisis Visual de Mapas Auto-organizativos. PhD Thesis. Madrid Politechnic University (2004)
23. Vesanto, J., Alhoniemi, E.: Clustering of the Self Organizing Map. IEEE Transactions on Neural Networks 11(3), 1082–1089 (2000)
24. Pölzlbauer, G.: Survey and Comparison of Quality Measures for Delf-Organizing Maps. In: Proceedings of the Fifth Workshop on Data Analysis, pp. 67–82 (2004)
25. Ahalt, C., Krishnamurthy, A.K., Chen, P., Melton, D.E.: Competitive learning algorithms for vector quantization. Neural Networks 3(3), 277–290 (1990)
26. Morales, V.P.: Análisis de Varianza para varias muestras independientes. Course Notes. Pontificia Comillas University (2011)

Identification of Different Types of Minority Class Examples in Imbalanced Data

Krystyna Napierala and Jerzy Stefanowski

Institute of Computing Sciences, Poznań University of Technology,
60–965 Poznań, Poland
{krystyna.napierala,jerzy.stefanowski}@cs.put.poznan.pl

Abstract. The characteristics of the minority class distribution in imbalanced data is studied. Four types of minority examples – safe, borderline, rare and outlier – are distinguished and analysed. We propose a new method for identification of these examples in the data, based on analysing the local neighbourhoods of examples. Its application to UCI imbalanced datasets shows that the minority class is often scattered without too many safe examples. This characteristics of data distributions is also confirmed by another analysis with Multidimensional Scaling visualization. We examine the influence of these types of examples on 6 different classifiers learned over various real-world datasets. Results of experiments show that the particular classifiers reveal different sensitivity to the type of examples.

Keywords: Imbalanced data, Classifiers, MDS Visualisation.

1 Introduction

Learning classifiers from imbalanced data has been receiving a growing research interest. Although several methods have already been introduced (see, e.g., a review in [2,4]), it is still worth asking a question about the nature of the class imbalance problem and about the properties of data distribution which make it so difficult. Some earlier studies, mainly based on experiments with artificial data, showed that simple class imbalance ratio was not the main difficulty. The degradation of classification performance is also related to other factors, e.g. to *decomposition* of the minority class into many sub-concepts with very few examples, which correspond to the *small disjuncts* [5]. Moreover, *overlapping* between classes strongly deteriorates the recognition of the minority class [3,9].

Following these related studies one could still look for other factors characterizing the data distribution. In our earlier papers [8] we hypothesized that some minority class examples could be located deeper inside the majority class. They could be treated as *outliers* or *rare cases* (if they are not single ones). We think that they should not be considered as a noise, as they are too rare and too precious for the minority class.

The role of the above mentioned data factors has been preliminary studied by us in the experiments with special artificial datasets [8]. Related research

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 139–150, 2012.

was also mainly focused on experimenting with artificial datasets [5,3,9]. By introducing a certain type of disturbance (e.g. overlapping or small disjuncts) and manipulating with its degree, the influence on the recognition of minority classes and on the abilities of particular classifiers were analysed.

In this study we direct our interest to the real-world imbalanced datasets. We would like to verify how often these factors actually occur in the data and to study their impact on the performance of different popular classifiers.

Our first aim is to analyse the distribution of examples in 19 real imbalanced datasets, mainly coming from the UCI repository[1] and often used in various experimental studies. We will show by analysing a 2D visualisation of a dataset obtained by *Multidimensional Scaling* (MDS) that in practice most of the datasets are seriously disturbed and that examples from the minority class can be of different nature. In our opinion, one can distinguish the following *types* of these examples: *safe*, *borderline*, *outliers* and *rare examples*.

The other aim of our study is to introduce a new method for identification of these types of examples in the data which is based on analysing a local neighbourhood of learning examples. In the experiments carried out with the same 19 datasets we plan to evaluate the amount of each type of examples. Depending on the main type of identified examples, we will categorize the datasets representing different characteristics of the minority class.

Finally, within each category of datasets, we compare the classification abilities of the classifiers – J48, PART, JRip, kNN, RBF and SVM. We want to verify whether they reveal different behaviour in face of different data types and how much they are sensitive to them.

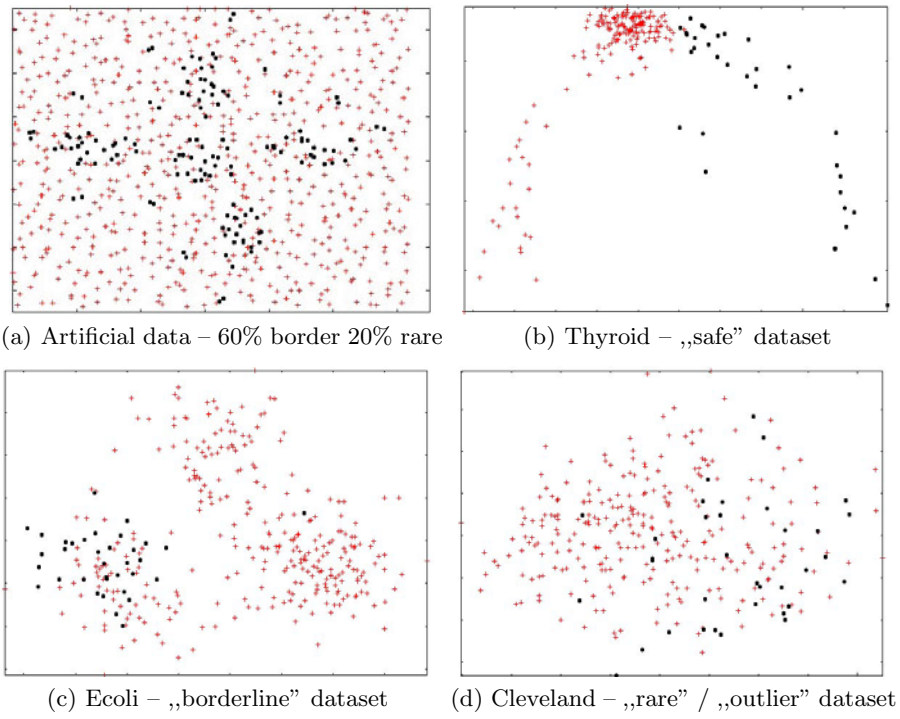
2 Distribution of Examples in the Minority Class

It is often claimed that learning from data with clearly separated classes is not difficult for most classifiers. It also concerns imbalanced data, as showed in the experimental studies, e.g. in [9]. Recognizing the minority class becomes more difficult when the distribution of examples from different classes is heavily mixed. Some researchers have also claimed that mutual position of examples has a crucial impact on learning from imbalanced data [3,6].

Several types of examples can be distinguished. The most common is the distinction between safe and unsafe examples [6]. *Safe* examples are located in the homogenous regions populated by the examples from one class only, otherwise they are treated as *unsafe* ones. Unsafe examples are often further discriminated between *borderline* and *noisy* examples as e.g. in [6]. *Borderline* examples are placed in the boundary regions between classes, where the examples from both classes overlap. Singular examples located deeper in the regions where the opposite class prevails, are usually treated as noisy examples. However, we share a different point of view. We claim that the minority class is often underrepresented in the dataset, so even the singular observations may represent a meaningful concept. What is more, as we will show in our experimental study, such

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html

examples often represent a considerable number of minority examples (even as much as half of the class). Therefore, we would like to pay special attention to these examples. If they are single examples surrounded by many examples from majority classes, we treat them as *outliers*. Although some of them may indeed be noisy observations, in general they are too precious to be automatically discarded. The observations distant from the core of the minority class may also form small groups of two-three examples. In such a situation they are even less likely to be noisy. We call them *rare examples*.



(a) Artificial data – 60% border 20% rare (b) Thyroid – ,,safe" dataset

(c) Ecoli – ,,borderline" dataset (d) Cleveland – ,,rare" / ,,outlier" dataset

Fig. 1. MDS visualisation of selected imbalanced datasets

To illustrate this categorization, in Fig. 1(a) we present an artificial dataset (coming from [8]) described with two numerical attributes, where the minority class (black circles) consists of different types of examples. It is divided into five sub-concepts (clusters). In each of these concepts, only the examples lying near the center of the cluster can be considered as *safe*. Many more examples belong to the border between the classes, in which the majority examples overlap with the minority ones. Finally, there are some examples more distant from the clusters, which could represent *outliers* or *rare examples*.

To confirm these observations in the real-world datasets, we visualise three commonly used imbalanced datasets from the UCI repository: thyroid, ecoli and

cleveland (Fig 1(b)-1(d)). As these datasets are described with more than two attributes, we use a *Multidimensional Scaling* technique (MDS) to reduce the dimensionality of the datasets. MDS performs a nonlinear mapping of dimensions with the aim of preserving the pairwise distances between data points in the original high dimensional data space into the projected low dimensional subspace [1]. As the datasets have both numeric and nominal attributes, we calculate the distances between the points using the HVDM metric [10].

Let us remark that using 2 dimensions in MDS requires keeping the data variance at a sufficient level. For instance, we could not use this technique to visualize the hepatitis dataset, as MDS with two dimensions preserved only 25% of variance in the dataset. For the other datasets, including three datasets visualised in Fig 1(b)-1(d), the percentage of preserved variance was higher than 60%, which in our opinion is enough to analyse the data.

Looking at (Fig 1(b)-1(d)) one can notice that the three data sets are of different nature. In thyroid dataset (Fig. 1(b)), the classes are clearly separated (even linearly), so most of the minority examples represent safe examples. In ecoli dataset (Fig. 1(c)) however, the classes seriously overlap. The consistent region belonging solely to the minority class (on the very left) is rather small – most examples lie in a mixed region between the classes. Finally, the cleveland dataset (Fig. 1(d)) is even more difficult to learn, as the minority class is very scattered – most examples form very small groups of few examples and some of the other are singular observations, surrounded by the opposite class. This dataset consists mostly of *rare examples* and *outliers*.

3 Assessing Types of Examples

Following the hypothesis about different types of examples in the minority class, we need an automatic procedure for their identification. We propose to assess the type of an example by analysing its local neighbourhood in the original attribute space[2]. For each minority example, we analyse the class assignment of its k-nearest neighbours. We use $k = 5$, because $k = 3$ may poorly distinguish the nature of examples, and 5 is often used in the preprocessing methods for class imbalance. With such k, the proportion of neighbours from the same class against neighbours from the opposite class can range from 5:0 (all neighbours are from the same class as the analysed example) to 0:5 (all neighbours belong to the opposite class). Depending on this proportion, we propose to assign the labels to the examples in the following way:

- 5:0 or 4:1 – an example is labelled as a safe example (further denoted as S).
- 3:2 or 2:3 – a borderline example (denoted as B). The examples with the proportion 3:2 are correctly classified by its neighbours, so they might still be safe. However, we prefer to be more pessimistic, and assume that they could be located too close to the decision boundary between the classes.

[2] The MDS projection to the reduced attribute space is applied for visualization aims only.

- 1:4 – labelled as a rare example (denoted as R), only if its neighbour from the same class has the proportion of neighbours either 0:5 or 1:4, but pointing to the analysed example. Otherwise there are some other examples from the same class in the proximity (although not in the immediate surrounding of $k = 5$), which suggests that it is rather a borderline example B.
- 0:5 – an example is labelled as an outlier and denoted as O.

To calculate the distance between examples we use the HVDM distance metric. It aggregates normalized Euclidean distances for numeric attributes with Stanfil and Valtz value difference metric for nominal attributes [10].

As our method is based on a simple analysis of a fixed number of neighbours, we want to check whether the assigned labels can precisely reflect the known distribution of examples. Inspired by good experience with artificial data in [8], we generated a number of such datasets (with 800 examples described by 2 numerical attributes) with varying imbalance ratios and number of the minority class sub-concepts, in which we changed the percentage of safe, borderline, rare and outlying examples. Table 1 presents the description of several analysed datasets and the labelling results.

Table 1. Labelling of artificial datasets

Dataset Description					Identified Labels			
Imbalance Ratio	Sub-concepts	Border [%]	Rare [%]	Outlier [%]	Safe [%]	Border [%]	Rare [%]	Outlier [%]
1:5	1	60	20	0	17.04	60.74	21.48	0.74
1:5	3	60	20	0	18.52	57.78	23.70	0.00
1:5	5	60	20	0	17.78	64.44	17.78	0.00
1:5	5	0	0	10	64.44	25.93	0.00	9.63
1:7	5	0	0	10	54.00	36.00	0.00	10.00
1:9	5	0	0	10	52.00	36.00	2.00	10.00

The first three datasets are disturbed in the same way (60% of borderline examples and 20% of rare examples), but differ in the number of sub-concepts. One of them (with 5 sub-concepts) is plotted in Fig. 1(a). Proportions of the identified labels show that our labelling method can correctly reconstruct the percentage of safe, borderline and rare examples, regardless of the number of sub-concepts. The other three datasets contain 10% of outliers and differ according to the imbalance ratio. Here, the labels also correctly reflect the percentage of outliers. However, although the classes in these datasets are not overlapped, a considerable number of examples is labelled as borderline. This is to some extent understandable, as the examples close to the border between the classes can contain in their neighbourhood some examples from the opposite class. Moreover, while labelling examples as borderline, we pessimistically assume that safe examples (3:2) also belong to this category.

4 Analysing Real-World Datasets

4.1 Datasets

We will conduct our analysis on 19 real-world datasets representing different domains, sizes and imbalance ratios. Their main characteristics are presented in the left-hand part of Table 2. 15 datasets come from the UCI repository and are often used in other works on class imbalance. Four datasets are retrospective medical datasets, which we used in our earlier works concerning imbalanced data[3]. If some datasets contain more than one majority class, we aggregate them into one class. The data are not modified, e.g. missing attribute values are handled directly by our methods and classifiers.

4.2 Labelling Results and Categorization of Datasets

The results of labelling the minority class examples in all the datasets are presented in the right-hand part of Table 2. The first observation is that most of the datasets contain the examples of all four types. Moreover, a majority of datasets contains rather a small number of safe examples. There are even such datasets as cleveland, glass, hsv or solar-flare, which do not contain any safe examples. Most of the data is characterized by a large number of difficult examples. Let us try to categorize considered datasets depending on the dominating type of examples from the minority class.

Only in abdominal-pain, acl, new-thyroid and vehicle datasets, safe minority examples prevail. Therefore, we can treat these 4 datasets as representatives of *safe* datasets (category S).

In the next category the borderline examples dominate in the distribution of the minority class. As could be observed in Table 1, even in datasets with clean borders a considerable amount of examples (up to 36%) can be labelled as borderline ones. So, the percentage of borderline examples must be even higher to represent some overlapping between classes. We treat a dataset as a *borderline* dataset if it contains more than 50% of B examples – these are credit-g, ecoli, haberman, hepatitis. Two additional datasets – car and scrotal-pain – are located somewhere between S and B categories. As the amount of safe examples is too low, we decide to assign them to the B category.

Then, several datasets contain many rare examples. Although they are not that numerous as B or S examples, they constitute even 20-30% of the minority class. The R category includes haberman (also assigned to B category), cmc, breast-cancer, cleveland, glass, hsv and abalone datasets, which have at least 20% of rare examples. Other datasets contain less than 10% of these examples.

Finally, some datasets contain a relatively high number of outlier examples – sometimes more than a half of the whole minority class. We assign the dataset to O category if more than 20% of examples are labelled as outliers. In Table 2

[3] We are grateful to prof. W.Michalowski and the MET Research Group from the University of Ottawa for abdominal-pain and scrotal-pain datasets; and to prof. K. Slowinski from Poznan University of Medical Science for hsv and acl datasets.

Table 2. Labelling of real-world datasets

Dataset Description				Identified Labels				
Dataset	Abbrev.	Size	Imbalance Ratio [%]	Safe [%]	Border [%]	Rare [%]	Outlier [%]	Type
abdominal-pain	AP	723	27.94	59.90	22.28	8.90	7.92	S
acl	AC	140	28.57	67.50	30.00	0.00	2.50	S
new-thyroid	NT	215	16.28	68.57	31.43	0.00	0.00	S
vehicle	VE	846	23.52	74.37	24.62	0.00	1.01	S
car	CA	1728	3.99	47.83	39.13	8.70	4.35	B
scrotal-pain	SP	201	29.35	38.98	45.76	10.17	5.08	B
credit-g	CG	1000	30	9.33	63.67	10.33	16.67	B
ecoli	EC	336	10.42	28.57	54.29	2.86	14.29	B
hepatitis	HE	155	20.65	15.63	62.50	6.25	15.63	B
haberman	HA	306	26.47	4.94	61.73	18.52	14.81	B, R
cmc	CM	1473	22.61	17.72	44.44	18.32	19.52	R
breast-cancer	BC	286	29.72	24.71	25.88	32.94	16.47	R
cleveland	CL	303	11.55	0.00	31.43	17.14	51.43	R, O
glass	GL	214	7.94	0.00	35.29	35.29	29.41	R, O
hsv	HS	122	11.48	0.00	0.00	28.57	71.43	R, O
abalone	AB	4177	8.02	8.36	20.60	20.60	50.45	R, O
solar-flare	SF	1066	4.03	0.00	48.84	11.63	39.53	O
transfusion	TR	748	23.8	18.54	47.19	11.24	23.03	O
yeast	YE	1484	3.44	5.88	47.06	7.84	39.22	O

these datasets are listed from cleveland to yeast. For many datasets, R and O categories appear together.

This categorization can be partly backed up by the MDS visualisation. The three datasets visualised in Fig. 1(b)-1(d) also show that new-thyroid is a safe dataset, ecoli can be assigned to a B category, while cleveland represents R and O categories.

5 Impact of Different Data Categories on Classifiers

The analysis of Table 2 showed that most datasets are seriously disturbed with a large number of B, R and O examples (or a mixture of them) which should cause difficulties in recognizing the minority class. Thus, in the next experiment we study the influence of these examples on the performance of popular classifiers.

We have decided to choose classifiers which are often used in related experimental studies and are based on different principles[4]. These are: decision tree learner J48 (a WEKA implementation of C4.5 classifier), two rule learners PART and Ripper (JRip), k-nearest neighbour (kNN), Naive Bayes, neural network (RBF) and SVM (SMO version). We parametrize them in the following way. J48 and PART are used without pruning. For JRip we do not change standard

[4] All implementation comes from WEKA platform.

options. kNN is used with $k = 1, 3, 5$ as we want to study whether increasing k influences the classifier. Naive Bayes is used with a supervised discretization of numeric attributes option from the WEKA's implementation. Standard values of parameters for RBF and SVM have failed to recognize the minority class. For RBF we have scanned several configurations trying to get the best sensitivity measures on all datasets. As a result, we changed a number of clusters to 5 and minimum standard deviation to 0.1. The similar optimization has been done for the SVM classifiers. We have used RBF kernel function, and selected two best combinations of complexity C and gamma G parameters – $(C = 50, G = 1.0)$, further referred to as SVM1 and $(C = 30, G = 0.1)$, denoted as SVM2.

Performance of the classifiers is evaluated with *Sensitivity* (true positive rate or an accuracy of the minority class), *Specificity* (accuracy of the majority class) and their aggregation by the geometric mean (*G-mean*) [4]. Their values are estimated by means of a 10-fold stratified cross-validation repeated 5 times to reduce possible variance. Table 3 presents the sensitivity and Table 4 – G-mean, with respect to 4 categories of datasets, which we will discuss below.

Table 3. Sensitivity of real-world datasets [%]

	DS	PART	J48	JRip	NB	1NN	3NN	5NN	RBF	SVM1	SVM2
S	**AP**	72.6	69.8	72.5	81.9	76.4	78.5	77.5	75.0	63.8	71.8
	AC	80.0	85.5	84.5	82.0	72.0	78.5	73.0	84.0	79.5	82.5
	NT	93.3	92.2	86.7	89.3	96.3	90.2	86.7	99.5	96.8	89.8
	VE	88.3	87.0	89.0	95.9	89.1	87.9	86.5	88.0	97.2	95.2
B	**CA**	90.0	77.7	47.0	0.0	3.1	3.1	3.1	49.6	27.0	88.2
	CG	47.7	46.5	37.6	50.5	50.3	39.9	37.1	43.6	2.5	52.2
	EC	42.0	58.0	59.7	81.0	52.2	50.8	57.8	54.7	64.0	58.5
	HA	33.4	41.0	34.0	25.0	30.1	26.9	18.1	18.3	14.7	1.3
	HE	45.7	43.2	31.2	75.5	44.0	37.0	47.5	60.7	39.3	51.5
	SP	63.4	55.3	53.4	56.5	58.4	58.7	49.2	62.5	32.0	65.9
R	**AB**	18.8	30.4	29.7	33.1	20.5	16.5	13.7	12.3	9.1	0.2
	BC	41.1	38.7	32.4	43.4	40.4	27.6	26.1	40.8	7.1	45.3
	CL	25.2	23.7	6.3	45.5	20.3	12.5	4.2	9.5	12.5	9.0
	CM	37.7	39.2	30.0	44.6	37.6	33.8	30.8	12.1	24.9	5.2
	GL	34.0	30.0	7.0	0.0	30.0	16.0	1.0	25.0	0.0	0.0
	HA	33.4	41.0	34.0	25.0	30.1	26.9	18.1	18.3	14.7	1.3
	HS	2.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
O	**AB**	18.8	30.4	29.7	33.1	20.5	16.5	13.7	12.3	9.1	0.2
	CL	25.2	23.7	6.3	45.5	20.3	12.5	4.2	9.5	12.5	9.0
	GL	34.0	30.0	7.0	0.0	30.0	16.0	1.0	25.0	0.0	0.0
	HS	2.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	SF	18.7	20.9	3.7	60.6	9.1	8.2	0.0	10.2	0.0	15.7
	TR	42.9	41.3	39.7	51.5	31.9	34.3	31.9	32.9	8.5	2.2
	YE	26.7	30.9	36.7	42.9	38.1	26.2	19.4	15.1	7.9	0.0

Category S
All classifiers can learn the minority class quite well – they recognize 70-90% of the minority examples, regardless of the used parameters. The G-mean values are also very high. It is difficult to appoint the best classifier. This confirms the results of the earlier experiments with artificial datasets, where two classes have been clearly separated, see e.g. [9].
Category B
Overlapping of examples causes more difficulty for the classifiers – they can usually recognize 30-50% of the minority class. However, the borderline examples influence classifiers in a different degree. Naive Bayes seems to work well on these datasets – it usually gives the highest sensitivity (except for car and haberman datasets). J48 and PART, which have quite similar learning strategies, also give very good results – often not as high as Naive Bayes, but more stable (they do not decrease on some data). The same refers to the RBF network. JRip also performs rather stable, but is usually worse than J48 and PART. For kNN, different settings of k result in a difference of about 10%, and it is difficult to say which values of k is the best. Generally, kNN works rather well on borderline datasets (with the exception of car dataset). SVM can achieve good results, but its performance depends on the used parameters. It seems that SVM2 is better for borderline datasets (apart from haberman dataset). By analysing Specificity and G-mean, we observed that overlapping affected less the majority class (Specificity values ranged between 80-90%); this is consistent with the conclusions from [3].
Category R
Datasets with many rare examples seem to be more difficult than borderline datasets – the average recognition of the minority class ranges between 0% and 40%. Again, Naive Bayes can give the best results, but it sometimes fails (e.g. on glass). One can also notice a more visible decrease of G-mean. Symbolic classifiers (PART, J48, JRip) perform relatively well and, what is important, their classification abilities are more stable (although JRip performs worse on glass and cleveland). KNN's performance is heavily sensitive to k. 1NN definitely dominates other configurations. We noticed that rare examples form groups of two/three examples, which can be correctly classified using one neighbour, while using $k = 5$ increases the probability of finding a majority neighbour, which often results in a negative prediction. At the same time, our analysis of Specificity measure showed that 1NN tends to degrade the performance in the majority class more than other classifiers, which is consistent with the observations from the experiments with artificial datasets conducted in [3]. However, as this degradation is not that serious (few percents), it does not impact the G–mean measure. Finally, SVM classifier is not suited for this kind of data. Although SVM1 seems better than SVM2 (contrary to B datasets), it is still worse than other classifiers. RBF gives unstable results, but it works better than SVM.
Category O
The results show that O type of data is definitely the most difficult for all classifiers. They usually cannot recognize more than 30% of the minority examples and they often cannot recognize any examples from this class (e.g. hsv

dataset). Nevertheless, it is still reasonable to use Naive Bayes, J4.8, PART or 1NN. 3NN, 5NN, RBF and both SVMs usually cannot learn the minority class. As for a majority class, all the classifiers can recognize it in a similar degree, reaching 95–100% on Specificity. So, the tendencies observed for Sensitivity are also demonstrated by values of the G-mean measure. This also indicates that for difficult data distributions, the classifiers are especially strongly biased toward the majority classes.

Table 4. G–mean of real-world datasets [%]

	DS	PART	J48	JRip	NB	1NN	3NN	5NN	RBF	SVM1	SVM2
	AP	78.6	78.1	80.0	85.7	79.8	82.6	82.8	82.6	76.8	79.9
S	**AC**	84.8	89.1	88.4	87.5	81.2	86.6	83.7	88.8	85.0	87.8
	NT	95.3	94.3	91.6	93.4	97.3	93.9	92.1	99.1	97.6	94.3
	VE	91.9	91.3	92.2	80.6	92.1	91.9	91.4	89.7	98.0	96.4
	CA	94.3	86.8	65.7	0.0	7.9	7.9	7.9	67.9	47.5	93.3
B	**CG**	60.2	59.1	56.7	65.7	63.7	58.1	56.9	61.0	11.5	65.2
	EC	55.4	69.2	70.9	81.7	66.8	66.3	70.1	65.7	74.8	71.1
	HA	46.8	53.8	47.4	33.9	44.6	43.9	33.4	34.4	31.0	3.1
	HE	54.9	53.9	43.8	79.6	56.1	51.5	61.5	71.9	52.7	64.7
	SP	70.7	67.2	63.0	70.5	68.7	72.3	66.1	74.0	51.2	74.7
	AB	41.9	53.9	53.2	55.3	43.2	38.8	35.8	32.2	28.2	1.4
R	**BC**	52.9	53.1	50.6	58.9	56.1	47.3	47.5	56.7	17.8	59.0
	CL	38.2	34.3	10.6	60.2	30.7	22.2	8.1	16.0	18.6	14.1
	CM	54.3	56.9	51.7	59.4	53.8	53.0	51.7	32.2	46.0	20.0
	GL	40.7	36.2	8.9	0.0	36.2	20.0	1.4	29.8	0.0	0.0
	HA	46.8	53.8	47.4	33.9	44.6	43.9	33.4	34.4	31.0	3.1
	HS	2.8	0.0	0.0	0.0	0.0	0.0	0.0	1.3	0.0	0.0
	AB	41.9	53.9	53.2	55.3	43.2	38.8	35.8	32.2	28.2	1.4
O	**CL**	38.2	34.3	10.6	60.2	30.7	22.2	8.1	16.0	18.6	14.1
	GL	40.7	36.2	8.9	0.0	36.2	20.0	1.4	29.8	0.0	0.0
	HS	2.8	0.0	0.0	0.0	0.0	0.0	0.0	1.3	0.0	0.0
	SF	31.9	37.6	6.4	73.8	17.8	16.6	0.0	18.8	0.0	26.8
	TR	60.2	59.9	58.8	64.6	50.2	53.9	52.9	54.4	25.7	8.6
	YE	42.0	49.7	56.9	59.7	58.3	43.8	34.1	27.1	17.7	0.0

The observed differences between classifiers could be analysed more precisely on a single dataset, by focusing attention on the classification errors made on the particular classified example. In other words, we want to analyse the distribution of error rates over types/labels of examples. To get their valid estimations, the dataset has to be big enough, to assure that there is a sufficient number of examples for all four labels. We choose one of the biggest datasets, abalone. In Table 5 we present the error rates for all labels. Let us observe that for each classifier, the error rate rises from the left to the right, confirming that most of the errors occur for the difficult types of examples. SVM, RBF, 3NN and 5NN cannot predict any of the rare and outlier examples. However, Naive Bayes and

Table 5. Error rates on labeled testing examples for abalone dataset [%]

Classifier	Safe	Border	Rare	Outlier
J48	9.29	48.16	75.51	82.72
PART	25.71	72.24	87.64	89.59
JRip	0.71	50.61	75.96	84.50
NB	0.00	38.37	70.79	84.14
1NN	18.57	54.69	77.53	97.87
3NN	6.43	56.33	95.06	98.11
5NN	13.57	62.04	97.75	99.29
RBF	17.86	64.08	99.33	100.00
SVM1	30.71	80.00	98.65	99.88
SVM2	100.00	100.00	99.10	100.00

a decision tree can recognize some of these examples. Most classifiers classify rather well the safe examples (but for SVM a choice of parameters is crucial), while in the borderline region all classifiers can recognize some of the examples.

6 Conclusions

Distribution of examples in the minority class and its influence on learning classifiers is the main topic of our study. We distinguish four types of examples – besides safe examples, we focus our attention on borderline, rare and outlier examples. The method for identification of these examples in the data is proposed, which is based on the analysis of the local neighbourhood of learning examples.

Our experiments with real-world datasets show that most datasets contain many unsafe examples. The minority class is usually decomposed or scattered, with only a small number of safe regions. This observation is confirmed by analysing a 2D visualisation of datasets obtained by Multidimensional Scaling.

Moreover, the distribution of the minority class can be of different nature – it may consist of borderline, rare or outlier examples. We categorize the datasets depending on the dominating type of examples and study the performance of different classifiers. Our experiments show that safe datasets are generally quite easy for all considered classifiers. Borderline and, even more, rare or outlier datasets, are a real source of difficulties and they influence classifiers in a different degree. We could also observe that the imbalance ratio and the size of the data are not as influential as the above distribution types. Comparison of the abilities of different classifiers shows that Naive Bayes and J4.8 trees or PART rules are the most robust to unsafe types of the minority class examples – also for more difficult types. Performance of kNN depends on the type of examples (works better for borderline and rare examples) and k=1 is usually a better value, but it can adversely affect the majority class. Then, RBF networks and SVM are quite sensitive to tuning parameters and fail to recognize rare or outliers examples.

Our observations are partly consistent with some earlier works with artificial datasets. In [3,5,8] it has also been shown that imbalance ratio is not the main

source of difficulty. In conclusions from [3], it has been suggested that when there is a large overlapping between the classes, SVM is significantly worse than any other algorithm when the minority class recognition is concerned, while 1NN tends to degrade the majority class more than other classifiers.

However, these earlier works do not attempt to analyse real-world datasets. They also do not generalize the observations on classifier's performance. We think that it is worth looking for methods able to evaluate the nature of real-world datasets and their degree of difficulty. Such analysis can help to foresee the behaviour of classifiers and their possible sensitivity to the type of examples which prevail in an analysed dataset. Besides our proposals of using labelling and MDS visualisation, other approaches could be developed – see e.g. some new techniques of data visualisation recently studied in [7].

Acknowledgments. The research has been supported by the Ministry of Science and Higher Education, grant no. N N519 441939.

References

1. Cox, T., Cox, M.: Multidimensional Scaling. Chapman and Hall (1994)
2. Fernández, A., García, S., Herrera, F.: Addressing the Classification with Imbalanced Data: Open Problems and New Challenges on Class Distribution. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS (LNAI), vol. 6678, pp. 1–10. Springer, Heidelberg (2011)
3. García, V., Sánchez, J., Mollineda, R.A.: An Empirical Study of the Behaviour of Classifiers on Imbalanced and Overlapped Data Sets. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 397–406. Springer, Heidelberg (2007)
4. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Data and Knowledge Engineering 21(9), 1263–1284 (2009)
5. Jo, T., Japkowicz, N.: Class Imbalances versus small disjuncts. ACM SIGKDD Explorations Newsletter 6(1), 40–49 (2004)
6. Kubat, M., Matwin, S.: Addresing the curse of imbalanced training sets: one-side selection. In: Proc. of Int. Conf. on Machine Learning ICML 1997, pp. 179–186 (1997)
7. Moreno-Torres, J.G., Herrera, F.: A Preliminary Study on Overlapping and Data Fracture in Imbalanced Domains by means of Genetic Programming-based Feature Extraction. In: Proc. of 10th Int. Conf. ISDA, pp. 501–506 (2010)
8. Napierała, K., Stefanowski, J., Wilk, S.: Learning from Imbalanced Data in Presence of Noisy and Borderline Examples. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS (LNAI), vol. 6086, pp. 158–167. Springer, Heidelberg (2010)
9. Prati, R., Batista, G., Monard, M.: Class imbalance versus class overlapping: an analysis of a learning system behavior. In: Proc. 3rd Mexican Int. Conf. on Artificial Intelligence, pp. 312–321 (2004)
10. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. J. Artif. Intell. Res. 6, 1–34 (1997)

Non-Disjoint Discretization for Aggregating One-Dependence Estimator Classifiers

Ana M. Martínez[1], Geoffrey I. Webb[2], M. Julia Flores[1], and José A. Gámez[1]

[1] Computer Systems Department, Intelligent Systems & Data Mining,
University of Castilla-La Mancha, Albacete, Spain
{anamaria.martinez,julia.flores,jose.gamez}@uclm.es
[2] Faculty of Information Technology,
Monash University, Melbourne, Australia
geoff.webb@monash.edu

Abstract. There is still lack of clarity about the best manner in which to handle numeric attributes when applying Bayesian network classifiers. Discretization methods entail an unavoidable loss of information. Nonetheless, a number of studies have shown that appropriate discretization can outperform straightforward use of common, but often unrealistic parametric distribution (e.g. Gaussian). Previous studies have shown the Averaged One-Dependence Estimators (AODE) classifier and its variant Hybrid AODE (HAODE, which deals with numeric and discrete variables) to be robust towards the discretization method applied. However, all the discretization techniques taken into account so far formed non-overlapping intervals for a numeric attribute. We argue that the idea of non-disjoint discretization, already justified in Naive Bayes classifiers, can also be profitably extended to AODE and HAODE, albeit with some variations; and our experimental results seem to support this hypothesis, specially for the latter.

Keywords: AODE, HAODE, Non-Disjoint Discretization, Bayesian Classifiers.

1 Introduction

So far, the AODE classifier [1] has arisen as one of the most attractive alternative to naive Bayes (NB), as it has proved to be significantly better in terms of error reduction compared to many others semi-naive techniques, maintaining under control its time and space complexity in training and classification time [2]. Nevertheless, as most of the techniques based on Bayesian networks, a multinomial probability distribution is assumed. This gives rise to difficulties in the context of continuous variables, as there is a need to infer joint probability distributions, and this is difficult in the absence of very large quantities of data. Two different ways to tackle this issue for AODE were studied in [3], that led to the Gaussian AODE classifier, where Conditional Gaussian networks were used to deal with datasets containing exclusively numeric attributes; and

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 151–162, 2012.

the Hybrid AODE classifier (HAODE), that resorted to the use of discretization only for the numeric values in the parents. However, these approaches also have their limitations, since the Gaussian assumption may be simply unrealistic. And discretization becomes a good alternative.

In this respect, we can find studies where the robustness of AODE and HAODE toward the discretization method is analyzed [4]. The conclusions in this work indicate that although the discretization method indeed matters when studying a particular dataset, it does not seem to be decisive when the aim is to compare a group of semi-naive Bayesian classifiers over a standard group of datasets. Nevertheless, only disjoint discretization techniques have been taken into account in that study. In [5], a novel non-disjoint discretization technique (NDD) is presented to cope with numeric attributes in NB by forming overlapping intervals. NDD forms overlapping intervals for a continuous attribute, always locating a value toward the middle of an interval to obtain more reliable probability estimations. Its use is based on the insight that while it is necessary to use a single discretization of each variable while classifying an instance, different discretizations can be applied when classifying different instances.

The results show a clear improvement in NB over other disjoint discretization methods, and we believe, that these results could also duplicate in AODE and HAODE, albeit with some modifications to the disjoint discretization method proposed. Compared to NB, AODE and HAODE could suffer more from creating a large number of intervals (from a variance increase), since their conditional probability tables (CPTs) are formed by the combination of a couple of attributes (the class and the parent). It is credible that NDD could help us to alleviate this problem by allowing larger intervals to be formed without greatly increasing the bias.

Hence, the main contributions of this paper are the following: to begin with, we redefine the original approach of NDD discretization for its use in AODE and HAODE, describing the corresponding modifications (Section 5). Furthermore, a new weighting system is included with the aim to decrease discretization bias. In Section 6, an experimental study compares the application of these *joint* discretization techniques in AODE and HAODE with the use of a traditional *disjoint* discretization method: equal frequency discretization (EFD)[1]. This study includes comparisons in terms of accuracy, but mainly focuses in results detailing bias and variance discretization records, as well as the combined error from both measures.

The rest of the paper is divided as follows: Section 2 and 3 introduces AODE and HAODE classifiers. Section 4 explains the main differences between disjoint and joint discretizations and finally, Section 7 provides our main conclusions from the study.

[1] As we will see below, this selection has not been made at random, but equal frequency division with 5 bins has shown to be the most beneficial for AODE [4].

2 AODE

AODE [1] is considered an improvement over NB and an interesting alternative to other attempts such as Lazy Bayesian Rules (LBR) [6] and Super-Parent TAN (SP-TAN) [7], since they offer similar error values, but AODE is significantly more efficient at classification time compared with the first one and at training time compared with the second one. In order to maintain efficiency, AODE is restricted to exclusively use 1-dependence estimators. Specifically, AODE can be considered as an ensemble of SPODEs (Superparent One-Dependence Estimators), because every attribute depends on the class and another shared attribute, designated as super-parent.

AODE computes the average of the n possible SPODE classifiers (one for each attribute in the database) and hence, the MAP (maximum a posteriori) hypothesis is as follows:

$$c_{MAP} = argmax_{c \in \Omega_C} \left(\sum_{j=1, N(x_j)>q}^{n} p(c, x_j) \prod_{i=1, i \neq j}^{n} p(x_i | c, x_j) \right), \qquad (1)$$

where x_i, x_j are the label of the predictive attributes and c the class label. The condition $N(x_j) > q$ is used as a threshold to avoid making predictions from attributes with few observations. In our experiments this q value has been set to 1, which is the default value in the data mining tool WEKA [8].

At *training time*, AODE has a $\mathcal{O}(mn^2)$ time complexity, where m is the number of training examples; whereas the space complexity is $\mathcal{O}(k(nv)^2)$, where v is the average number of values per attribute and k the number of classes. The resulting time complexity at *classification time* is $\mathcal{O}(kn^2)$, while the space complexity is $\mathcal{O}(k(nv)^2)$.

3 HAODE

NB can deal with hybrid (discrete and numeric variables) datasets by means of Gaussian and multinomial distributions. On the contrary, this is not possible for AODE, as a numeric variable (super-parent) cannot be the parent of a discrete variable. This is the reason why AODE can only be applied after discretizing numeric variables. HAODE [3] restricts the use of discretization to only the variable which acts as super-parent in every model, keeping it numeric when it is playing the role of *child*. Thus, multinomial distributions are estimated for discrete variables and the super-parent, together with one univariate Gaussian distribution (one for each configuration in the Cartesian product between the class and the super-parent) for each numeric variable which acts as child.

Classification is performed according to the following equation then:

$$c_{MAP} = argmax_{c \in \Omega_C} \left(\sum_{j=1, N(x_j)>m}^{n} p(x_j, c) \prod_{i=1 \wedge i \neq j}^{n} \mathcal{N}(x_i : \mu_i(c, x_j), \sigma_i^2(c, x_j)) \right), \qquad (2)$$

where $\mu_i(c, x_j)$ and $\sigma_i^2(c, x_j)$ are the mean and variance of X_i conditioned to the values c for the class and x_j for X_j. $\mathcal{N}(x_i : ., .)$ is the resulting value of the normal density function of x_i with the corresponding mean and variance.

HAODE presents the same time complexity as AODE, but achieves a slight reduction in spatial complexity because HAODE requires only two parameters (mean and variance) for Gaussian distributions, independently of the number of states in which this variable has been discretized when it acts as super-parent.

4 Disjoint vs. Non-Disjoint Discretization

Formally, given the numeric attribute values $x_i, x_j \in \mathbb{R}$, any disjoint discretization method would create a unique interval $(a, b] \ni x_i$ and $(d, e] \ni x_j$ for every value so that AODE's statistics, $p(X_j = x_j, C = c)$ and $p(X_i = x_i | C = c, X_j = x_j)$ would be estimated by

$$p(X_j = x_j, C = c) \approx p(d < X_j \leq e, C = c) \tag{3}$$

$$p(X_i = x_i | C = c, X_j = x_j) \approx p(a < X_i \leq b | C = c, d < X_j \leq e) \tag{4}$$

In disjoint discretization techniques (EFD, equal width division, MDL, etc) every numeric sample belongs to a single interval. I.e., considering $x_i < x_j$, if $a \neq d$ (they do not fall in the same interval) then $d \geq b$. This implies that for those cases where the original numeric value falls around the center of the interval assigned, we could expect more distinguishing information than when it falls near one of the boundaries of the interval.

In contrast, NDD creates bins that overlap. So long as a single bin is used consistently when classifying a single object, it does not matter whether inconsistent bins are used when classifying different objects.

4.1 Equal Frequency Discretization

EFD is an unsupervised technique where the values are ordered and divided into b disjoint bins so that each one contains approximately the same number of training instances.

Therefore, every bin contains m/b instances with adjacent values, where m is the total number of samples. This type of discretization method provides bins containing equal numbers of examples and hence the variance of the estimates formed from the bins should be more stable than alternatives. As a group of values with identical values must be placed in the same bin, it is not always possible to generate b intervals with exactly the same number of values.

Time complexity for this technique is $\mathcal{O}(m \log m)$ as it is necessary to perform an ordering of the data.

4.2 Non-Disjoint Discretization

NDD is also an unsupervised technique that forms t atomic intervals $B_0 = [a'_1, b'_1]$, $B_1 = (a'_2, b'_2], \ldots, B_t = (a'_t, b'_t]$ (where $b'_i = a'_{i+1}, \forall i$), with equal frequency. In its definition for NB [5], one operational interval or label is formed

then for each set of three consecutive atomic intervals, such that the rth ($1 \leq r \leq t-2$) interval $(a_r, b_r]$ satisfies $a_r = a'_r$ and $b_r = b'_{r+2}$. Each numeric value x is assigned to interval $(a'_{i-1}, b'_{i+1}]$ where i is the index of the atomic interval $(a'_i, b'_i]$ such that $a'_i < x \leq b'_i$, except when $i = 1$ in which it is assigned to interval $(a'_1, b'_3]$ and when $i = t$ that it is assigned to interval $(a'_{t-2}, b'_t]$. Here t and the number of instances per atomic interval are selected proportionally to the number of training instances, following the idea of Proportional k-Interval Discretization [9].

NDD is dominated by sorting as well, and hence, its complexity is also $\mathcal{O}(m \log m)$.

5 NDD Adapted to AODE and HAODE

By dividing the ranges of numeric attributes into overlapping intervals in AODE and HAODE, we not only intend to reduce discretization bias [10] by always locating a value toward the middle of an interval and, in general, creating a larger number of intervals; but also maintaining discretization variance, since the number of samples from which the CPTs will be estimated should be similar.

Intuitively, discretization resulting in large interval numbers tends to have low bias (any given interval is less likely to include a decision boundary of the original numeric attribute). Discretization resulting in intervals with a large number of instances tends to have low variance (as the probability estimations are more stable and reliable). The problem is that supposing there is a fixed dataset size, the larger the number of intervals, the smaller the number of instances per interval is.

The application of NDD to AODE involves discretizing the whole dataset into non-disjoint intervals before training the classifier, whereas in the case of HAODE, just the cases where a numeric attribute plays the role of super-parent will be discretized.

However, and for the reasons that we detail next, some changes are introduced to the original definition of NDD as specified in Yang and Webb's paper [5]:

1. A threshold is considered to mark the minimum frequency from which an atomic interval will not be merged with its neighbours. This should prevent us from increasing bias when sufficient samples are already provided. See Figure 1 for an example on interval formation having each atomic interval frequency into account. Since it is possible the presence of multiple instances with the same value, the number of final samples per atomic attribute may vary, and it usually does[2].
2. In the original definition of NDD, the interval size is equal to the interval number ($\approx \sqrt{m}$) with the aim to give equal importance to discretization bias and discretization variance reduction. Even though it provides very good results for NB, it is not the case for AODE or HAODE, where in general, a

[2] The way in which this is handled is the same for NDD and EF5, check WEKA's equal frequency discretization method for more details.

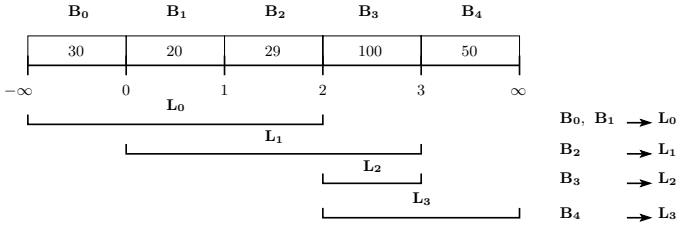


Fig. 1. Example of NDD division, the minimum frequency to merge atomic intervals into a single label (L) is equal to 100. Labels selected when classifying samples belonging to atomic bins B_0, B_1, B_2, B_3 and B_4 are indicated at the bottom right corner.

smaller number of intervals is desired because it is necessary to estimate the probability of an interval on one attribute conditioned by both an interval on another attribute and the class, whereas in NB it is necessary only to estimate the probability of an interval given the class. Previous experiments have shown that Proportional Discretization (PD), tailored to NB, where a number of $\sqrt{m}$ instances is selected, is not generally beneficial for AODE [4].

3. When the number of cut-points is lower than 3, then equal frequency discretization will be kept.
4. Weighting importance: note that by using NDD as defined above, there are some numeric samples that fall within two or three labels. Given a numeric sample x_i discretized by NDD into the labels $L_1 = (a'_1, b'_1]$, $L_2 = (a'_2, b'_2]$ and $L_3 = (a'_3, b'_3]$ in training time; L_2 would be the final label assigned to another sample $x_j \in \mathbb{R}$, $x_j = x_i$ in classification time. The contribution of L_2 to the CPT will be greater (it is given more importance when training) than the contribution provided by the other two bins. This is carried out by the use of weights. There exist several forms in which these weights could be distributed, in this first approach we have adopted the simplest one (apart from uniform distribution, being equivalent to non-weighting). Since a single sample can be allocated at most in three atomic bins, the weight distribution could be set as 0.75 for the centred label and the rest equally divided into the other labels (if there is more than one)[3]. In AODE, the combination of weights when both the parent and the child involved in a CPT come from a joint discretization is carried out by multiplying its corresponding weights (so that the sum remains equal to one).

   Figure 2 shows an example for a training instance I with two numeric attributes: X_0 and X_1. This instance is discretized using the NDD procedure indicated in Section 5, obtaining I_{NDD}. Hence, the value 3.5 for X_0 falls within three labels: L_0, L_1 and L_2 (specifically centred in L_1, that is why it is given the highest weight), whereas the value 2 for X_1 falls

[3] Further experiments have been carried out by slightly altering the weight assignment obtaining very similar results. This study has been performed using 3 atomic bins per interval, and we believe that this result may not be extrapolated to any higher odd number.

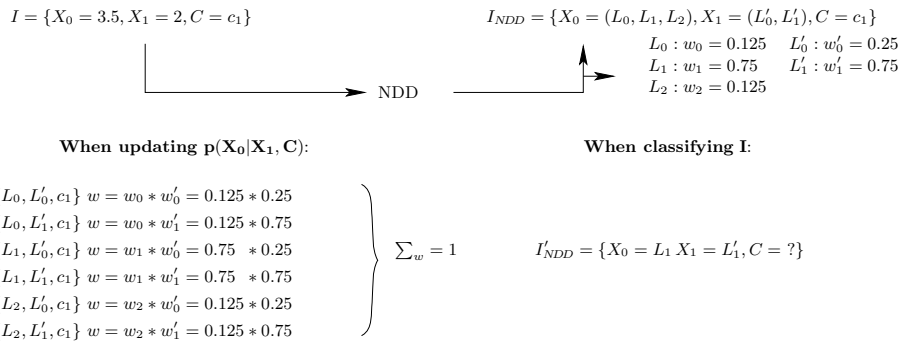


Fig. 2. Example on how weighted NDD works in AODE: first of all, the instance is discretized using NDD and weights are assigned to every label. When training, instance I would contribute to the CPT for X_0 given X_1 and C as shown in the left hand side. If classifying I, only the main labels (so that the sample is in the center) are considered.

within labels L'_0 and L'_1 (centred in L'_1 in this case). These weights are then used to indicate the contribution of each pair of values when updating the CPTs[4]. When the same instance I were to be classified (the class is missing), then I'_{NDD} would be used, where the centred labels for both attributes are considered. Then the MAP equation would be the following: $argmax_{c \in \Omega_C} \left(\sum_{j=1, N(x_j)>q}^{n} p(c, L^*_j) \prod_{i=1, i \neq j}^{n} p(L^*_i | c, L^*_j) \right)$, where L^*_j, is the centred label for X_j; and L^*_i, the centred label for X_i.

As NDD is dominated by sorting, no increase in the complexity is induced.

6 Experiments

We run our experiments on 28 datasets from the UCI machine learning repository [11] and KDD archive [12], listed in Table 1. As in [5], this experimental suite comprises 3 parts. The first part is composed of all the UCI datasets used by [13] when publishing the entropy minimization heuristic discretization. The second part is composed of all the datasets with numeric attributes used by [14] for studying NB classification. The third part is composed of larger datasets employed in [9].

To begin with, we have pre-processed the datasets using an unsupervised filter to replace all the missing values with the modes and means from the existing data in the corresponding column, and another one to remove useless attributes that do not vary at all or whose variation percentage is lower than 99%[5]. This is

[4] Note that in a multinomial distribution, the combination of values from an instance to be incorporated in a CPT contribute with a `unit`, whereas here we consider the contribution of the `weight` for each label (that always sums to one for each instance).

[5] These two filters have been applied with the default settings provided by WEKA.

Table 1. Main characteristics of the datasets: number of predictive numeric variables (n), number of predictive discrete variables (d), number of classes (k) and number of instances (m)

Id	Datasets	n	d	k	m	Id	Datasets	n	d	k	m
1	labor-negotiations	8	8	2	57	15	annealing	6	32	6	898
2	echocardiogram	5	1	2	74	16	german	7	13	2	1000
3	iris	4	0	3	150	17	multiple-features	3	3	10	2000
4	hepatitis	6	13	2	155	18	hypothyroid	7	18	2	2163
5	wine-recognition	13	0	3	178	19	satimage	36	0	6	6435
6	sonar	60	0	2	208	20	musk	166	0	2	6598
7	glass-identification	9	0	3	214	21	pioneer-mobile-robot	29	7	57	9150
8	heart-disease	7	6	2	270	22	handwritten-digits	16	0	10	10992
9	liver-disorders	6	0	2	345	23	sign-language	8	0	3	12546
10	ionosphere	34	0	2	351	24	letter-recognition	16	0	26	20000
11	horse-colic	7	14	2	368	25	adult	6	8	2	48842
12	credit-screening	6	9	2	690	26	impums.la.99	20	40	13	88443
13	prima-indians-diabetes	8	0	2	768	27	census-income	8	33	2	299285
14	vehicle	18	0	4	846	28	forest-covertype	10	44	7	581012

in order to make the group of datasets uniform and suitable for all the classifiers considered in the comparison.

In order to evaluate the experimental results we use two methods: accuracy (for the sake of comparison with previous works and to facilitate the frame to reproduce experiments) and error in terms of bias and variance according to [15], using five times 2-fold cross validation (5x2cv). 5x2cv entails a reasonable trade-off between precision and execution time of the experiments, providing a better partition for the posterior statistical analysis, as in addition, the degree of overlapping between the different folds is lower [16]. The bias-variance decomposition has been performed using the sub-sampled cross-validation procedure as specified by [17]. The global error obtained by this procedure is the sum of the bias and variance results.

The discretization technique selected as the basis for comparison is EFD with 5 bins (EF5), as it has shown to provide slightly better results compared to other methods [4] such as Minimum Description Length [13], equal width discretization or equal frequency discretization with a different number of bins.

As advanced in section 4, the labels formed in NDD will comprise at most 3 atomic bins[6]. To provide a fair comparison with EF5, the initial number of atomic bins considered is 15. This means that the labels created (groups of three atomic bins) will be of approximately the same average size as the bins for EF5. The minimum frequency from which an atomic interval will not be merged with its neighbours will be 100 (approximately 30 per atomic bin[7]).

[6] In theory any odd number would be acceptable (the larger the better to allocate a sample in the middle of an interval), but for simplicity we take 3 as in [5].

[7] The figure 30 has been selected motivated by the 30-sample rule-of-thumb very recurrent in statistics. Still, further experiments were carried out with different values; although the results were not significantly different, the best values were obtained with 30 and $33.\overline{3}$.

Table 2. Results in terms of `accuracy±sample standard deviation` obtained for AODE and HAODE using EF5, NDD and NDDw

	AODE			HAODE		
Id	EF5	NDD	NDDw	EF5	NDD	NDDw
1	93.3333±3.88	•94.3860±4.49	•94.0351±5.35	90.8772±8.59	•91.2281±9.76	•91.2281±9.48
2	68.9189±4.81	•72.9730±4.77	•72.1622±7.10	74.3243±4.64	72.9730±5.41	•76.7568±5.44
3	92.9333±2.74	•93.8667±2.20	•93.0667±1.97	95.8667±1.83	•96.0000±2.08	95.4667±1.80
4	82.1935±2.81	•82.9677±3.86	•82.4516±3.54	83.2258±2.23	82.0645±3.23	82.7097±3.09
5	96.4045±1.28	•96.8539±1.66	○96.4045±1.48	98.0899±0.93	○98.0899±0.76	97.8652±0.98
6	81.4423±3.63	•81.6346±4.19	80.7692±4.03	82.7885±3.91	82.5000±3.76	•84.6154±3.74
7	68.1308±5.07	•68.5047±4.06	•70.2804±3.76	69.1589±4.13	•69.5327±4.83	•70.0000±4.77
8	81.4815±2.42	•83.4815±2.59	•81.7037±1.60	81.0370±1.95	•81.5556±1.96	○81.0370±2.84
9	60.3478±3.47	•65.1014±3.46	•63.5942±2.76	62.0290±3.56	59.5942±3.81	61.1014±3.40
10	91.3390±2.19	89.4017±2.55	90.3134±2.42	92.2507±2.33	•92.7066±1.68	•92.4217±1.37
11	79.5652±1.23	•80.1087±1.94	•80.7609±1.18	65.6522±4.65	•66.3043±4.42	•66.4674±3.95
12	86.4638±1.23	•86.5797±0.95	•86.5507±1.18	80.7826±1.16	•80.0870±0.88	80.0870±1.14
13	75.2083±1.78	•75.5208±1.33	74.1927±1.65	75.6250±0.90	75.2344±0.94	75.0260±1.08
14	69.2199±1.14	68.3215±1.39	67.9433±1.58	73.3806±2.05	•73.5225±2.13	72.2695±2.25
15	87.9955±1.77	86.3474±1.65	•90.0668±1.07	82.9176±1.61	81.9822±2.81	82.5167±2.64
16	74.1600±1.08	•74.3800±0.93	•74.3400±1.19	73.7400±1.27	•74.7800±1.16	•74.1800±1.10
17	66.2600±1.22	•68.1700±1.26	•68.3600±1.31	69.1800±1.37	•69.9400±1.68	•70.6800±1.60
18	97.3000±0.21	•98.1979±0.27	•98.2548±0.22	98.1284±0.23	•98.3181±0.26	•98.3307±0.33
19	87.4219±0.57	•88.4444±0.40	•88.4444±0.40	83.9254±0.98	•85.9176±0.79	•85.9176±0.78
20	85.2743±0.85	•93.2404±0.32	•93.2555±0.30	83.5920±1.09	•87.5750±0.71	•87.5720±0.71
21	90.5268±0.47	•93.5432±0.86	•93.5016±0.87	89.1607±0.86	•94.3607±0.67	•94.3497±0.67
22	97.0287±0.17	96.8013±0.32	96.8013±0.32	97.1634±0.33	•97.6638±0.21	•97.6638±0.21
23	71.3678±0.70	•73.2680±0.51	•73.2855±0.51	66.3399±1.01	•67.1433±0.86	•67.1242±0.88
24	83.4580±0.21	•85.4120±0.37	•85.4720±0.37	84.5250±0.21	•88.1870±0.32	•88.2030±0.32
25	83.9347±0.25	•84.1677±0.29	•84.2771±0.29	84.0830±0.31	•83.9237±0.37	83.9892±0.35
26	92.3890±0.08	92.3854±0.08	•92.3928±0.08	87.0904±0.44	•87.7243±0.58	•87.7017±0.57
27	92.1766±0.09	•92.4165±0.07	•92.4171±0.07	93.4646±0.11	•93.6628±0.09	•93.6666±0.09
28	71.3988±0.11	•73.9682±0.09	•73.9682±0.09	69.9027±0.13	•70.8710±0.09	•70.8710±0.09
Av.	82.4169±1.62	•85.5873±1.67	•83.5381±1.67	81.7251±1.89	•82.2658±2.01	•82.4935±1.99

Table 2 shows the accuracy results obtained for AODE and HAODE using EF5, NDD and NDDw; along with the sample standard deviation for each dataset. The bullet next to certain outputs (in NDD and NDDw) indicates that the corresponding result improves the output provided when EF5 is used. The circle, in turn, indicates a draw. These results lead us to think that the use of NDD or NDDw is competitive over EF5 (and by extension, other traditional disjoint discretization techniques), especially for the former. Nevertheless, standard deviation is, on average, higher for NDD and NDDw compared to EF5, this indicates that the latter is more robust with respect to the income data, although the values provided in terms of accuracy are lower, in spite of that.

Table 3 shows the number of datasets for which discretizing with NDD obtained better, equal or worse performance compared to using equal frequency with 5 bins. These records are complemented by the results from the Wilcoxon signed-rank tests [18], which compare every pair of algorithms considering the whole group of datasets. The first two columns depict the records when the samples are not weighted (i.e. weighted uniformly) according to the atomic bin to which they belong. In this case, NDD in AODE and HAODE is better at improving accuracy and global error. The improvement is clear also as far as bias is concerned for HAODE and variance for AODE. However, this advantage is not as clear in terms of bias in AODE and variance in HAODE, although they still

Table 3. Comparisons in terms of win-draw-lose records and Wilcoxon tests

	non-weighted		weighted	
w-t-l / Wilcoxon	AODE NDD vs EF5	HAODE NDD vs EF5	AODE NDDw vs EF5	HAODE NDDw vs EF5
Accuracy	23-0-5 / **< 0.05**	21-1-6 / **< 0.05**	22-1-5 / **< 0.05**	18-2-8 / **< 0.05**
Bias	14-3-11 / 0.2395	21-1-6 / **< 0.05**	15-3-10 / **< 0.1**(0.06)	22-0-6 / **< 0.05**
Variance	18-2-8 / **< 0.05**	14-4-10 / 0.3621	13-2-13 / 0.6	10-0-14 / 0.863
Error	21-1-6 / **< 0.05**	19-3-6 / **< 0.05**	16-3-9 / **< 0.05**	18-1-9 / **< 0.05**

Table 4. Average results in terms of accuracy/bias/variance/error (best value in bold)

	AODE		HAODE	
	EF5	NDD	EF5	NDD
Accuracy	82.4169	**83.5873**	81.7251	**82.2658**
Bias	0.1298	**0.1250**	0.1348	**0.1275**
Variance	0.0395	**0.0355**	0.0440	**0.0435**
Error	0.1737	**0.1643**	0.1836	**0.1758**

provide better records compared to EF5, no statistical difference is found. If we consider the weighted version of NDD, the results are slightly better in terms of bias (specially for AODE), at the expense of variance and overall worsening. Hence, from now on in the paper, we will just consider non-weighted NDD, although it is important to observe that the increase in variance may have less effect on error when larger data are provided.

Table 4 displays the average results in terms of accuracy, bias, variance and error obtained for the different classifiers, where NDD outperforms in every pair-to-pair comparison.

Note that execution time comparisons would show no interest information, since differences are minimum (same complexity order).

Hence, in the light of these results one question arises: why does NDD seems to improve more pronouncedly AODE's variance and HAODE's bias compared to applying equal frequency? The difference between the two classifiers lies in the "double use" (in parents and children nodes) of NDD in AODE, which seems to help in reducing variance at the expense of a bias sacrifice.

In this study, even though there is a slight improvement of HAODE over AODE (16-0-12 in terms of accuracy, see Table 2), this is not as striking as in the original study in 2009 [3], and this difference even shifts to 13-1-14 when NDD is applied. We believe this fact might be motivated by two reasons:

- HAODE aims to avoid information loss by resorting to the use of discretization only when necessary for the super-parents. However, that implies that Gaussian distributions are assumed in some cases, which can be a handicap if the real distributions in data are not Gaussians.
- In general, we should prefer high-bias, low-variance classifiers when the data are sparse; and low-bias, high-variance classifiers when data are numerous. Since we are now dealing with larger datasets, we could also deduce that

HAODE is more robust in small ones and AODE in larger ones, unless the normality condition is satisfied.

7 Conclusions

In this paper, we have studied the impact of applying NDD to AODE and HAODE compared to traditional disjoint discretization techniques. In this study we have chosen equal frequency division to represent the latter, as it was shown previously to provide better results among the most common disjoint discretization methods (EF, equal width division, MDL, etc).

We have introduced some modifications to the original definition of NDD [5] in order to fit into AODE and HAODE's context, as a smaller number of bins is usually desired compared to NB to avoid increasing variance. Furthermore, a new weighting system has been introduced at the counting process in order to increase the importance given to the bins created by NDD where samples are placed in the middle; which provided better results in terms of bias but worse overall records.

The results have been analyzed in terms of accuracy, bias, variance and global error obtaining the following conclusions:

- In general terms, an overall improvement is found for the two classifiers (AODE and HAODE) when NDD is used. Statistical differences according to the Wilcoxon test are found for both classifiers as far as accuracy and global error (sum of bias and variance) is concerned.
- The analysis on error descomposition in terms of bias and variance displays better results at all times when using NDD, being this improvement more marked for HAODE in terms of bias, and AODE in terms of variance.

The most important conclusion though, is the fact that whereas some of the most common disjoint discretization techniques have failed to demonstrate consistent improvement relative to alternatives, non-disjoint discretization demonstrates better win/draw/loss records and significant overall improvement. Still, we plan to extend the experimental part to a test bed of high dimensional datasets in order to corroborate these conclusions.

Moreover, we believe that the positive results observed in AODE are a good motivation to think that the beneficial properties of NDD will be strengthen when applied to Aggregating n-dependence estimators (AnDE) [19], for values of n greater or equal to 2 (since when $n = 1$ it is equivalent to AODE).

One drawback of NDD is that it requires the user to select additional parameters apart from the number of bins to form (such as in equal frequency division), also the number of atomic bins per operational interval and the minimum frequency per interval must be chosen.

Acknowledgements. This work has been partially funded by FEDER funds, the Spanish Government (MICINN) and the Castilla-La Mancha regional Government (JCCM) through projects TIN2010-20900-C04-03 and PEII11-0100-7773,

the FPU grant with reference number AP2007-02736 and the Australian Research Council under grant DP110101427.

References

1. Webb, G.I., Boughton, J.R., Wang, Z.: Not So Naive Bayes: Aggregating One-Dependence Estimators. Mach. Learn. 58(1), 5–24 (2005)
2. Zheng, F., Webb, G.I.: A Comparative Study of Semi-naive Bayes Methods in Classification Learning. In: Simoff, S.J., Williams, G.J., Galloway, J., Kolyshkina, I. (eds.) Proc. of the 4th AusDM Conf., pp. 141–156 (2005)
3. Flores, M.J., Gámez, J.A., Martínez, A.M., Puerta, J.M.: GAODE and HAODE: two proposals based on AODE to deal with continuous variables. In: Danyluk, A.P., Bottou, L., Littman, M.L. (eds.) ICML. ACM Int. Conf. Proc. Series, vol. 382, p. 40. ACM (2009)
4. Flores, M.J., Gámez, J.A., Martínez, A.M., Puerta, J.M.: Handling numeric attributes when comparing bayesian network classifiers: does the discretization method matter? Appl. Intell. 34(3), 372–385 (2011)
5. Yang, Y., Webb, G.I.: Non-disjoint discretization for naive-bayes classifiers. In: Sammut, C., Hoffmann, A. (eds.) Proc. of the 9th Int. Conf. on Mach. Learn (ICML 2002), pp. 666–673. Morgan Kaufmann, San Francisco (2002)
6. Zheng, Z., Webb, G.I.: Lazy Learning of Bayesian Rules. Mach. Learn. 41(1), 53–84 (2000)
7. Keogh, E.J., Pazzani, M.J.: Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In: Proc. of the 7th Int. Workshop on AI and Statistics, pp. 225–230 (1999)
8. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann (2005)
9. Yang, Y., Webb, G.I.: Proportional k-Interval Discretization for Naive-Bayes Classifiers. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 564–575. Springer, Heidelberg (2001)
10. Yang, Y., Webb, G.I.: Discretization for Naive-Bayes Learning: Managing Discretization Bias and Variance. Mach. Learn. 74(1), 39–74 (2009)
11. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
12. Hettich, S., Bay, S.D.: The UCI KDD Archive (1999), http://kdd.ics.uci.edu
13. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuousvalued attributes for classification learning. In: 13th Int. Joint Conf. on AI, vol. 2, pp. 1022–1027. Morgan Kaufmann (1993)
14. Domingos, P., Pazzani, M.J.: On the optimality of the simple bayesian classifier under zero-one loss. Mach. Learn. 29(2-3), 103–130 (1997)
15. Kohavi, R., Wolpert, D.H.: Bias plus variance decomposition for zero-one loss functions. In: Proc. of the 13th Int. Mach. Learn., pp. 275–283 (1996)
16. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. 10, 1895–1923 (1998)
17. Webb, G.I., Conilione, P.: Estimating bias and variance from data (2002)
18. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (2006)
19. Webb, G.I., Boughton, J., Zheng, F., Ting, K.M., Salem, H.: Learning by extrapolation from marginal to full-multivariate probability distributions: decreasingly naive Bayesian classification. Machine Learning (in-press)

An Adaptive Hybrid and Cluster-Based Model for Speeding Up the *k*-NN Classifier

Stefanos Ougiaroglou[1,⋆], Georgios Evangelidis[1], and Dimitris A. Dervos[2]

[1] Dept. of Applied Informatics, University of Macedonia, 54006 Thessaloniki, Greece
{stoug,gevan}@uom.gr
[2] Dept. of Informatics, Alexander TEI of Thessaloniki, 57400 Sindos, Greece
dad@it.teithe.gr

Abstract. A well known classification method is the k-Nearest Neighbors (k-NN) classifier. However, sequentially searching for the nearest neighbors in large datasets downgrades its performance because of the high computational cost involved. This paper proposes a cluster-based classification model for speeding up the k-NN classifier. The model aims to reduce the cost as much as possible and to maintain the classification accuracy at a high level. It consists of a simple data structure and a hybrid, adaptive algorithm that accesses this structure. Initially, a preprocessing clustering procedure builds the data structure. Then, the proposed algorithm, based on user-defined acceptance criteria, attempts to classify an incoming item using the nearest cluster centroids. Upon failure, the incoming item is classified by searching for the k nearest neighbors within specific clusters. The proposed approach was tested on five real life datasets. The results show that it can be used either to achieve a high accuracy with gains in cost or to reduce the cost at a minimum level with slightly lower accuracy.

Keywords: *k*-NN classifier, cluster-based classification, data reduction.

1 Introduction

The *k*-Nearest Neighbors (*k*-NN) classification algorithm is a widely-used lazy classifier [2]. When a new item needs to be classified, *k*-NN searches the available data (training set) and retrieves the *k* nearest neighbors to it according to a distance metric (e.g., euclidean distance). The new item is classified to the class that is the most common one among the classes of the retrieved *k* nearest neighbors, with possible ties resolved either randomly or by choosing the class of the nearest neighbor.

The *k*-NN classifier is considered to be an effective classifier and has many applications. Its main drawback is that the computational cost needed to compute all distances between a new item and the training data can be very high. The research conducted on the reduction of the *k*-NN cost is based on indexing, data reduction and cluster-based methods.

⋆ S. Ougiaroglou is supported by the State Scholarship Foundation of Greece (I.K.Y.).

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 163–175, 2012.

Multi-attribute indexing methods can speed up the nearest neighbor searching procedures [13]. They are very effective when datasets have moderate dimensionality (e.g. 2-10). In higher dimensions, the phenomenon of "dimensionality curse" renders those indexes irrelevant since their performance degrades rapidly and can become worse than that of the sequential scan.

Data Reduction Techniques (DRTs) [16,5,9,17,15] build a small representative set of the initial data, often called Condensing Set (CS). The idea is to apply k-NN on this set attempting to achieve almost the same accuracy as with the original data at a much lower cost. These methods can be divided into two main categories: (i) filtering (or selection) [5], and (ii) abstraction (or generation) [16] algorithms. Filtering algorithms select items from the Training Set (TRS) as representatives. On the other hand, abstraction algorithms generate representatives by summarizing similar items.

Finally, Cluster-Based Methods (CBM) preprocess the TRS items and place them into clusters [8,19]. When a new item must be classified, they dynamically decide which subset of the training data will be used. Contrary to DRTs, CBMs and indexing methods do not reduce the storage requirements.

Our motivation is to address the problem of classifying large and high-dimensional datasets, where dimensionality reduction negatively affects the accuracy, thus, indexing is not applicable. We combine two strategies, abstraction and clustering, in a hybrid classification schema to speed-up the k-NN classifier. First, a clustering preprocessing task builds a two level data structure. Its first level contains cluster centroids (representatives) for each class, and, its second level contains the set of items belonging to each such cluster. Then, an adaptive and hybrid algorithm attempts to achieve high accuracy while keeping the computational cost as low as possible. A new item is classified either using the first or the second level of the data structure. So, an abstraction and a cluster-based approach are combined to achieve the desirable performance.

The rest of this paper is organized as follows. Section 2 briefly presents the related work, and Section 3 considers in detail the proposed classification model. In Section 4, experimental results based on real life datasets are presented, and the paper concludes in Section 5.

2 Related Work

One of the most widely used filtering algorithms is the Condensing Nearest Neighbor (CNN) rule [6]. It reduces the cost of k-NN by removing the non close-border items. The idea is that these items can be removed without significant loss of accuracy. The CNN-rule determines the amount of the selected representative items automatically based on the level of noise in the data and the number of classes. Many other algorithms either extend the CNN-rule or are based on the same idea. However, the CNN-rule algorithm continues to be the reference algorithm and it is used in many works for comparison purposes.

A recently proposed filtering algorithm is the Prototype Selection by Clustering (PSC) [12]. PSC is based on the idea that homogeneous clusters

(not containing items that belong to different classes) include items that lie in the "internal" area of a class, whereas, non-homogeneous clusters include close-border items. PSC uses k-means clustering in order to divide the TRS into clusters (any clustering method can be used). Then, for each homogeneous cluster, the nearest item to the cluster centroid is placed into the CS, whereas, for each non-homogeneous cluster, only the items that define the decision boundaries are placed into the CS.

Chen and Jozwik proposed a well-known abstraction algorithm [1]. Chen's algorithm is based on dividing the TRS into small subsets. The algorithm begins by finding the pair of the most distant points, A, B, in the TRS. It continues by splitting the TRS into two subsets. One subset contains the items nearest to A, and, the other the items nearest to B. Then, it selects to split the subset with the greatest diameter. This procedure continues until the number of subsets becomes equal to the user-predefined CS size. Finally, Chen's algorithm computes a mean item for each subset. The class of each mean item is defined to be the most common class in the corresponding subset.

Sanchez introduced three Reduction by Space Partitioning (RSP) algorithms that are based on the idea of Chen's algorithm [14]. Contrary to Chen's algorithm, RSP1 computes as many mean items as the number of different classes in each subset. RSP1 and RSP2 differ on the way that they select the subset that will be divided. RSP3 continues splitting until all subsets are homogeneous. Thus, RSP3 determines the CS size automatically.

Editing approaches constitute a subcategory of filtering algorithms. Their goal is to increase the accuracy rather than reduce the cost. This is achieved by removing noisy and close-border data, leaving smoother decision boundaries. The reduction rates of many filtering/abstraction algorithms depend on the level of noise that exists in the data. Thus, in many classification tasks, an editing algorithm is used to remove the noisy data before the application of the main reduction procedure [9]. However, some hybrid filtering approaches, such as the DROP algorithms [17], integrate the idea of editing. They build the CS and simultaneously remove noisy items (see [5] for details). A well known editing algorithm is the Edited Nearest Neighbor (ENN) rule [18]. For each TRS item x, if the class of x does not agree with the majority of its k nearest neighbors, x is removed. ENN-rule needs to compute $\frac{N*(N-1)}{2}$ distances, i.e., all the distances among the TRS items.

Many other Abstraction and Filtering algorithms are reviewed, categorized, evaluated and compared to each other in [16] and [5] respectively. Other relevant reviews can be found in [9,15,17].

Hwang and Cho have proposed an effective CBM [8]. It uses the k-means algorithm [11] to find clusters in the data. Then, each cluster is divided into two sets. Items located in a certain distance from the cluster centroid are placed into the "core set", while the rest are placed into the "peripheral set". If a new item lies within the "core area" of the nearest cluster, it is classified by retrieving the k-nearest neighbors from this cluster. Otherwise, the nearest neighbors are retrieved from the set formed by the items of the nearest cluster and the

"peripheral" items of adjacent clusters. Another effective CBM is the Cluster-based Tree [19]. It is based on searching in a cluster hierarchy and can be used for either metric or non-metric spaces.

3 The Proposed Classification Model

The proposed model consists of two parts: (i) a Speed-Up Data Structure (SUDS) built by a clustering preprocessing procedure, and, (ii) a Hybrid and Adaptive Classification Algorithm that uses this structure.

3.1 Speed-Up Data Structure Construction

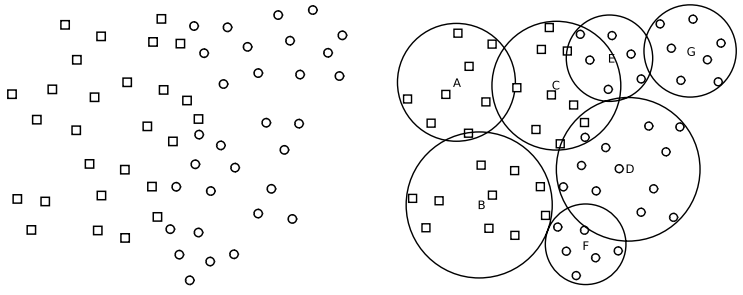
Initially, the training data is preprocessed by the k-means algorithm to form the data structure (SUDS). More specifically, for each class, k-means identifies a number of clusters. SUDS consists of two data levels. The first level is a list of representatives (the mean vectors of all clusters for all classes). Each node of this list points to a list that contains the items assigned to the specific representative. These lists of items form the second level of SUDS.

We use a parameter, the Data Reduction Factor (DRF), to determine the number of clusters that will be created. For each class C, the number of clusters NC is estimated by $NC = \lceil \frac{|C|}{DRF} \rceil$, where $|C|$ is the number of items that belong to C. Thus, the DRF parameter specifies the number of the clusters that will be created. Figure 1 illustrates a two-dimensional example. In this case there are two classes, square and circle. The initial dataset includes 27 squares and 31 circles (Figure 1(a)). Thus, if DRF is set to 10, the classes Square and Circle should be represented by 3 and 4 mean vectors respectively (Figure 1(b)). The result of the clustering procedure will be the two level SUDS depicted in Figure 1(c). Class square is represented by the mean vectors A–C, and class circle by D–G.

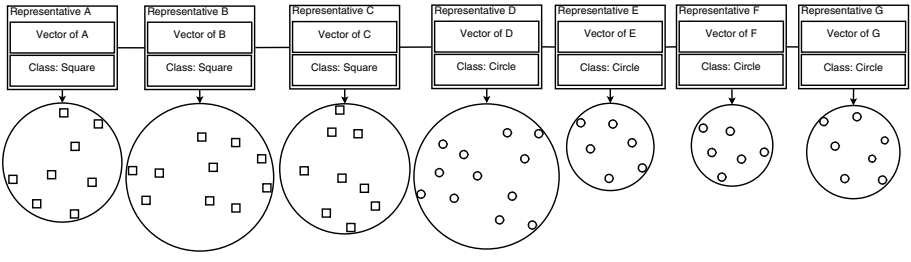
The idea of creating multiple class representatives via clustering has also been proposed by Datta and Kibler in [3]. Their goal was the representation of distant and disjoint groups formed by items of the same class and the building of a classifier able to manage symbolic (nominal) attributes. Also, Hruschka et al in [7] have proposed an imputation method (missing values completion) that also uses the k-means algorithm on the data of each class.

3.2 The Fast Hybrid and Adaptive Classification Algorithm

The second part of the model comprises a Fast Hybrid and Adaptive Classification Algorithm (FHACA) that accesses the SUDS (Algorithm 1). FHACA uses three (input) parameters that let the user define the desirable trade-off between accuracy and cost. The idea behind the algorithm is quite simple. When a new item x has to be classified, FHACA initially scans the first level of SUDS (first level search) and retrieves the Rk nearest representatives to x. If the acceptance criterion introduced by the $NRRatio$ parameter is met, these representatives determine the class where x belongs to. Upon failure, x is classified by searching



(a) Initial dataset (b) Clustered dataset



(c) Speed-up data structure

Fig. 1. k-means clustering on items of each class (DRF=10)

for the k "real" nearest neighbors within the clusters of the Rk nearest representatives (second level search). Obviously, the more items classified without the need of the second level search, the lower is the computational cost involved.

FHACA uses the $NRRatio$ parameter to decide when to switch to a second level search. This parameter defines how many nearest representatives should determine the majority class (the most common class among the Rk nearest representatives) in order to classify the new item. For instance, suppose that the input parameters are set to be k=3, Rk=10, and $NRRatio$=70%. Also, suppose that a new item x should be classified and a SUDS with 100 representatives is available. FHACA, initially, retrieves and examines the 10 nearest representatives from the first level of SUDS. If 7 or more of them belong to the majority class, then x is classified to this class. Otherwise, the 3 "real" nearest neighbors are retrieved from the data subset formed by the union of clusters (second level of SUDS) of the 10 nearest representatives, and they determine the class of x. Even in the case of the second level search, FHACA avoids searching in the the rest 90 clusters.

3.3 Discussion

Considering the proposed classification algorithm, it is obvious that an unclassified item that lies in a close-border area, is classified by a second level search. On the other hand, an item that lies in the "internal" area of a class, is classified

Algorithm 1. The Fast Hybrd and Adaptive Classification Algorithm

Input: *SUDS, k, Rk, NRRatio*

```
1: for each unclassified item x do
2:    Scan 1st level of SUDS and retrieve the Rk Nearest Representatives (NRs) to x
3:    Find the majority class MC1 of the Rk NRs (ties are resolved by 1-NR)
4:    MCCounter ← COUNT(representatives of the majority class)
5:    if MCCounter ≥ NRRatio then
6:       Classify x to MC1
7:    else
8:       Scan within the set formed by the union of the clusters of the Rk representa-
         tives and retrieve the k Nearest Neighbors (NNs) to x
9:       Find the majority class MC2 of the k NNs (ties are resolved by 1-NN)
10:      Classify x to MC2
11:   end if
12: end for
```

by the mean vectors (first level search). Thus, the proposed method is neither a cluster-based nor an abstraction method, since it dynamically decides on how a new item is classified. The first level comprises an abstraction method that uses the mean vectors obtained by the clustering preprocessing procedure. On the other hand, the second level comprises a cluster-based method that uses a dynamically-formed subset of the initial TRS. Hence, the method is a hybrid approach. Moreover, it differs from the DRTs in that, as in the cases of indexing and CBMs, it does not reduce the storage requirements.

Concerning the parameters of the proposed model, Rk and $NRRatio$ should be determined by taking into account the DRF value that was used for SUDS construction. If accuracy is more critical than cost and a SUDS with few and large clusters is available, Rk and $NRRatio$ should have high values. On the other hand, if cost is more critical and a SUDS with many and small clusters is available, low Rk and $NRRatio$ values are recommended. Considering the DRF value, low DRF values are recommended for building accurate classifiers with high cost savings and high DRF values for building fast classifiers without significant accuracy loss. If our needs are not specified at the time that SUDS is constructed, an intermediate DRF value is the most appropriate. In this case, the trade-off can be afterwards determined by adjusting Rk and $NRRatio$.

Furthermore, when FHACA executes a second level search, it accesses a subset of the initial TRS formed by the union of the Rk clusters. Since each cluster includes items of a specific class, this dataset is an almost noise free dataset (it does not include noisy items of classes which are not represented by the Rk representatives) and, thus, the classification accuracy is not affected as much by noisy data. Taking into account this property, editing is not as necessary as in many filtering/abstraction algorithms. Of course, noise removal and overlapping "cleaning" among regions of different classes could increase the cluster quality and the overall classification performance.

Table 1. Dataset description (cost is in million distance computations)

Dataset	Train/Test dataset size	Attr.	Classes	Best k	Accuracy (%)	Cost (M)
Letter Recognition(LR)	15000/5000	16	26	4	95.68	75
Magic Gamma Telescope(MG)	14000/5020	10	2	12	81.39	70.28
Pendigits(PD)	7494/3498	16	10	4	97.89	26.21
Landsat Satellite(LS)	4435/2000	36	6	4	90.75	8.87
Shuttle(SH)	43500/14500	9	7	1	99.88	630.75

4 Performance Evaluation

The proposed model was implemented in C and evaluated using five real life datasets distributed by the UCI Repository [4]. They are summarized in Table 1. All datasets were used without data normalization or any other transformation.

The cost measurements were estimated by counting the distance computations required to classify all items of the Testing Set (TES) by scanning the training data. All distances were estimated using the Euclidean distance metric. Since the conventional k-NN classifier (conv-k-NN) requires the computation of all distances between each TES item and the initial TRS items, its cost can be estimated by multiplying the cardinalities of the training and testing sets (see the second and last column of Table 1). For instance, conv-k-NN computes $15000 \times 5000 = 75$M distances for LR. The fifth column lists the k value found to achieve the highest accuracy. For comparison purposes, we implemented in C two filtering, an abstraction, and, a cluster-based approach. We selected CNN-rule [6], PSC [12], RSP3 [14] and Hwang's algorithm [8], respectively.

4.1 Experimental Setup

The adaptive schema of the proposed model provides four parameters: DRF, Rk, $NRRatio$, and, k. We defined k to be the best k value of conv-k-NN (Table 1). Several experiments were conducted for the other parameters. The values tested for each one were: (a) DRF 4,6,8,10,20,30,...,300, (b) Rk 1,2,...,30, and (c) $NRRatio$ 51%, 70% and 100%. Thus, for each dataset, we built and evaluated 2970 ($33 \times 30 \times 3$) FHACA classifiers. In the end, we kept the most accurate FHACA classifier for each reported cost. In real life applications, there is no need to do such extensive tests to determine the appropriate values of the parameters. Here, our purpose was to fully understand how each parameter influences the model construction and its performance. In real life applications, the parameters should be determined by taking into consideration the accuracy and cost significance as well as the dataset used.

Hwang's algorithm also uses four parameters: C is the number of clusters, L is the number of adjacent clusters, D is the threshold that defines the core and peripheral items, and, k defines the number of nearest neighbors (see Section 2).

We set $L = \lfloor\sqrt{k}\rfloor$ as Hwang and Cho did in their experiments. We set $C = \lfloor\sqrt{\frac{n}{2^i}}\rfloor$, i=1,...,8, where n is the number of items. Thus, for each dataset, we built 8 classifiers. The first classifier (for i=1) is based on the rule of thumb that defines $C = \lfloor\sqrt{\frac{n}{2}}\rfloor$ [10]. We decided to build classifiers that use small C values based on the observation that Hwang and Cho defined C=10 for a TRS with 60919 items (they did not find the optimal C value). Of course, the fewer and larger the clusters, the higher the cost of the classifiers (see Fig. 2-6). Following the approach of Hwang and Cho, we considered as peripheral items, those whose distance from the cluster centroid was greater than the double average distance among the items of each cluster (i.e. D=2). Finally, we chose the k values that achieved the highest accuracy.

Another issue that needs attention is the number of clusters that PSC uses. In [12], the authors executed experiments by constructing $r \times j$, $j = 2, 4, \cdots, 10$, clusters, where r is the number of discrete classes in a dataset. Since our main goal is to achieve high accuracy at a low cost, we decided to test PSC with higher j values. We conducted several experiments with varying j values (up to 200 and for the noisy MG dataset up to 2000).

Concerning CNN-rule, PSC and RSP3, two experiments were conducted for each one, one on the original and one on the edited TRS. For editing purposes, we implemented the ENN-rule [18] and used it by setting k=1 for all datasets. Thus, each method was tested with two k-NN classifiers, one for each condensing set. We refer to them as CNN-kNN, ENN-CNN-kNN, RSP3-kNN, ENN-RSP3-kNN, PSC-kNN and ENN-PSC-kNN. The k parameter for classifiers was adjusted to achieve the highest accuracy.

Since only the first level search (FHACA-1st LS) can be used to classify new items, we present its performance in the diagrams of subsection 4.2. FHACA-1st LS carries out the whole task when $NRRatio$ is set to zero. In other words, the k-NN classifier classifies the new data using only the set of representatives produced by the k-means algorithm. As in the case of FHACA, we kept the most accurate FHACA-1st LS and PSC classifiers for each reported cost.

4.2 Comparisons

Table 2 presents a small subset of preprocessing measurements. Specifically, it includes the number of representatives/clusters as well as the preprocessing costs required by each method (the cost of editing is not included). The very high preprocessing cost of RSP3 is the result of the farthest point computations in the subsets. Since the preprocessing is executed only once, these cost measurements may be not so significant. However, they have to be evaluated taking into account the performance that the corresponding classifiers achieve.

We now focus on the accuracy-cost measurements obtained by executing the classifiers on the five datasets. In the following, we will be reporting the FHACA parameter values in parenthesis in this order (DRF, Rk, $NRRatio$). Figures 2 through 6 report, for each classifier, the cost measurements on the x-axis and the corresponding accuracy values on the y-axis.

Table 2. Preprocessing

Method		LR	MG	PEN	LS	SH
CNN-rule	Items:	2517	5689	312	909	300
	Cost:	145,386,010	217,900,759	7,940,953	13,545,272	57,958,973
RSP3	Items:	5906	6646	857	1219	688
	Cost:	291,151,380	412,752,916	70,561,629	28,929,950	15,671,718,080
PSC	Items.:	3075	4024	362	689	591
r=10	Cost:	187,371,957	17,796,445	20,254,707	6,970,603	259,638,105
PSC	Items.:	2813	3868	582	746	633
r=50	Cost:	429,018,237	122,380,296	86,182,235	30,608,866	1,027,702,841
PSC	Items.:	3583	3838	1056	1000	664
r=100	Cost:	390,007,005	187,853,770	112,410,384	42,579,896	1,777,429,582
Hwang	Clust.:	86	83	61	47	147
i=1	Cost:	56,778,655	235,903,403	18,294,684	21,683,796	556,375,731
Hwang	Clust.:	10	10	7	5	18
i=7	Cost:	11,715,045	11,634,045	951,759	691,870	36,061,653
SUDS	Clust.:	3769	3502	1880	1112	10880
DRF=4	Cost:	12,832,249	243,430,728	9,922,935	6,679,461	4,372,102,123
SUDS	Clust.:	1515	1402	755	447	4353
DRF=10	Cost:	8,907,558	273,815,604	6,220,083	4,876,310	2,910,000,719
SUDS	Clust.:	763	702	380	224	2178
DRF=20	Cost:	5,585,990	161,018,546	4,356,282	2,976,219	1,929,776,967
SUDS	Clust.:	389	352	192	113	1091
DRF=40	Cost:	3,569,682	87,023,904	3,558,287	2,377,213	2,032,629,026
SUDS	Clust.:	160	141	80	47	440
DRF=100	Cost:	1,520,838	51,632,480	1,553,272	1,285,263	1,462,204,134
SUDS	Clust.:	82	71	40	26	222
DRF=200	Cost:	671,965	26,615,055	368,204	478,202	972,749,420

Letter Recognition (Figure 2): FHACA achieved accuracy between 94%–96% with the lowest cost. For instance, an effective FHACA classifier achieved an accuracy of 95.52% with 4.1M computations (30,10,100). All other methods had higher cost and achieved lower accuracy. The first three Hwang classifiers (i=1–3) may be preferable for accuracy levels between 93%–94%. The highest FHACA accuracy was 96% (8,27,100).

Magic Gamma Telescope (Figure 3): Again, FHACA achieved high accuracy (over 81%) having lower cost than all other methods. For instance, FHACA achieved an accuracy of 81,39% requiring 6.26M computations (40,19,100). A "cheeper" FHACA classifier required 2.62M computations for an accuracy of 81.14% (50,5,100). The highest FHACA accuracy (81.47%) was achieved with 10M computations (210,7,100). CNN-kNN, PSC-kNN and RSP3-kNN were affected by the high level of noise that exists in this dataset. ENN-rule managed to remove many noisy items and consequently, CNN-rule, PSC and RSP3 achieved higher reduction rates when they executed on edited data.

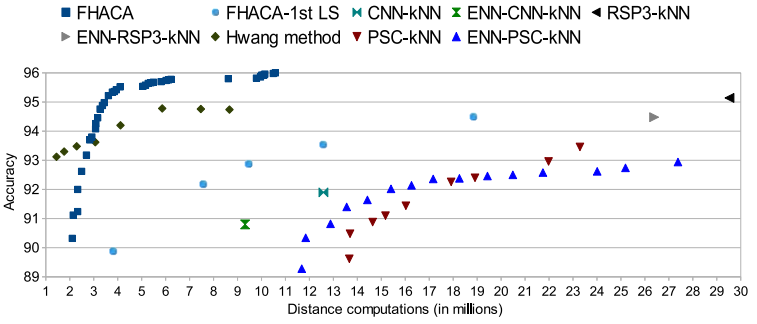


Fig. 2. Letter Recognition Dataset

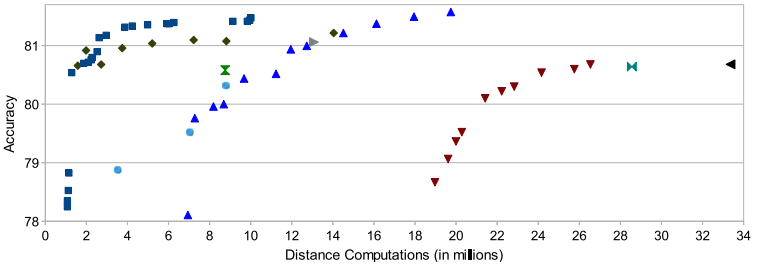


Fig. 3. Magic Gamma Telescope Dataset

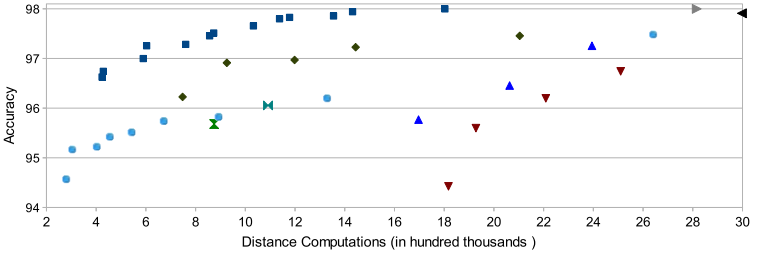


Fig. 4. Pendigits Dataset

Pendigits (Figure 4): FHACA had the best performance. It achieved accuracy values between 96.6%–98%. In all cases, it outperformed Hwang's algorithm. FHACA and RSP3 achieved higher accuracy than that of the conv-k-NN. The most accurate (98%) FHACA classifier required 1.8M computations (50,10,100). A faster FHACA classifier required about 0.6M computations and achieved an accuracy of 97.26% (80,3,70).

Landsat Satellite (Figure 5): Once again, FHACA performed very well. An accuracy of 90.75% was achieved with 533,282 computations (50,5,100). A FHACA classifier with half of that cost achieved an accuracy of over 90%

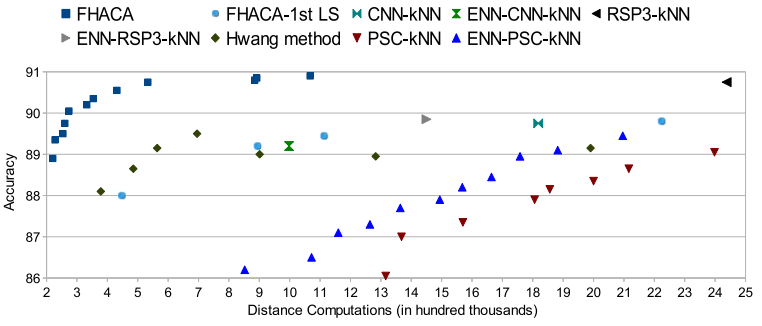


Fig. 5. Landsat Satellite Dataset

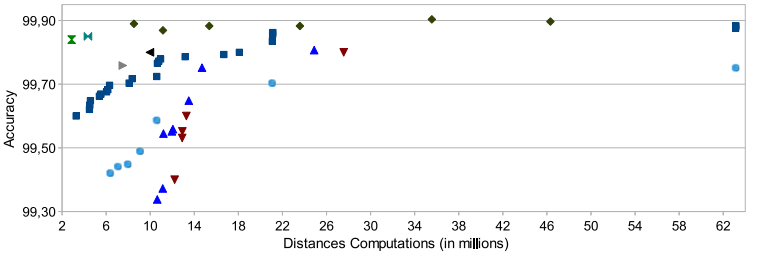


Fig. 6. Shuttle Dataset

(40,2,51/70/100). The highest FHACA accuracy was 90.9% and the corresponding cost was 1.067M (70,7,100). RSP3-kNN achieved an accuracy of 90.75, at a much higher cost. The other methods did not achieve accuracies above 90%.

Shuttle (Figure 6): Shuttle is an imbalanced dataset. Approximately 80% of the data belongs to one class. The two CNN approaches performed very well. This happened because the large class forms a "clear" and "tight" cluster and so, CNN-rule successfully removed a huge amount of data. The performance of RSP3 was close to that of CNN. Hwang's algorithm achieved the highest accuracy value. Compared to the results of the previous four datasets, FHACA had worse perfomance. Although it achieved high accuracies (even 99,883%), it required high cost. This is because FHACA constructs many non-necessary representatives for the majority class. Nevertheless, FHACA performed comparably to the other methods when it classified test items belonging to rare classes.

Considering the experimental results on all datasets, we can conclude that the proposed model can achieve comparable or higher accuracy at a lower cost than the other methods.

5 Conclusion and Future Work

We proposed a cluster-based model for speeding-up the k-NN classifier. The model involves the construction of a two level data structure and an algorithm

that makes predictions using either the first or the second level of this structure. The model lets the user define the desirable trade-off between accuracy and cost. Thus, it can be used either to improve the accuracy with gains in cost, or to reduce the cost at the minimum level without sacrificing accuracy. Experimental results showed that significant improvement may be achieved, with the accuracy remaining at high levels and comparable to that of the conventional k-NN.

We plan to incorporate in our method a mechanism for dynamically adapting the *NRRatio* parameter in relation to the number of representatives of each class. The main goal of this extension is to efficiently deal with imbalanced datasets. In addition, we will combine our method with abstraction/filtering approaches and we will devise algorithms for the dynamic updating of SUDS.

References

1. Chen, C.H., Jóźwik, A.: A sample set condensation algorithm for the class sensitive artificial neural network. Pattern Recogn. Lett. 17, 819–823 (1996)
2. Dasarathy, B.V.: Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society Press (1991)
3. Datta, P., Kibler, D.: Learning symbolic prototypes. In: Proceedings of the Fourteenth ICML, pp. 158–166. Morgan Kaufmann (1997)
4. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
5. Garcia, S., Derrac, J., Cano, J., Herrera, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Transactions on Pattern Analysis and Machine Intelligence 99(prePrints) (2011)
6. Hart, P.E.: The condensed nearest neighbor rule. IEEE Transactions on Information Theory 14(3), 515–516 (1968)
7. Hruschka, E.R., Hruschka, E.R.J., Ebecken, N.F.: Towards efficient imputation by nearest-neighbors: A clustering-based approach. In: Australian Conference on Artificial Intelligence, pp. 513–525 (2004)
8. Hwang, S., Cho, S.: Clustering-Based Reference Set Reduction for k-nearest Neighbor. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4492, pp. 880–888. Springer, Heidelberg (2007)
9. Lozano, M.: Data Reduction Techniques in Classification processes (Phd Thesis). Universitat Jaume I (2007)
10. Mardia, K., Kent, J., Bibby, J.: Multivariate Analysis. Academic Press (1979)
11. McQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proc. of 5th Berkeley Symp. on Math. Statistics and Probability, pp. 281–298. University of California Press, Berkeley (1967)
12. Olvera-Lopez, J.A., Carrasco-Ochoa, J.A., Trinidad, J.F.M.: A new fast prototype selection method based on clustering. Pattern Anal. Appl. 13(2), 131–141 (2010)
13. Samet, H.: Foundations of multidimensional and metric data structures. The Morgan Kaufmann series in computer graphics. Elsevier,Morgan Kaufmann (2006)
14. Sánchez, J.S.: High training set size reduction by space partitioning and prototype abstraction. Pattern Recognition 37(7), 1561–1564 (2004)
15. Toussaint, G.: Proximity graphs for nearest neighbor decision rules: Recent progress. In: 34th Symposium on the INTERFACE, pp. 17–20 (2002)

16. Triguero, I., Derrac, J., García, S., Herrera, F.: A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE Transactions on Systems, Man, and Cybernetics, Part C 42(1), 86–100 (2012)
17. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning 38(3), 257–286 (2000)
18. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. on Systems, Man, and Cybernetics 2(3), 408–421 (1972)
19. Zhang, B., Srihari, S.N.: Fast k-nearest neighbor classification using cluster-based trees. IEEE Trans. Pattern Anal. Mach. Intell. 26(4), 525–528 (2004)

A Co-evolutionary Framework for Nearest Neighbor Enhancement: Combining Instance and Feature Weighting with Instance Selection

Joaquín Derrac[1], Isaac Triguero[1], Salvador García[2], and Francisco Herrera[1]

[1] Dept. of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, 18071 Granada, Spain
{jderrac,triguero,herrera}@decsai.ugr.es
[2] Dept. of Computer Science, University of Jaén, 23071 Jaén, Spain
sglopez@ujaen.es

Abstract. The nearest neighbor rule is one of the most representative methods in data mining. In recent years, a great amount of proposals have arisen for improving its performance. Among them, instance selection is highlighted due to its capabilities for improving the accuracy of the classifier and its efficiency simultaneously, by editing noise and reducing considerably the size of the training set. It is also possible to remark the role of feature and instance weighting as outstanding methodologies for improving further the performance of the nearest neighbor rule.

In this work we present a new co-evolutionary algorithm for combining the former techniques. Its performance is compared with evolutionary approaches performing instance selection, instance weighting and feature weighting in isolation, as well as with the nearest neighbor classifier. The results obtained, contrasted through nonparametric statistical tests, supports the capabilities of co-evolution as a outstanding strategy for joining several proposals for enhancing the nearest neighbor rule.

Keywords: Co-evolution, Instance Selection, Instance Weighting, Feature Weighting, Evolutionary Algorithms, Nearest Neighbor Classifier.

1 Introduction

The k-nearest neighbor classifier (k-NN) is one of the best known techniques in data mining. It is one of the most used algorithms in supervised classification. Due to its simplicity, effectiveness and precision, it has attracted a great interest by the research community [16].

Instance selection is a well-known proposal for improving the performance of the k-NN classifier [10,8]. Its application allows us to reduce the spatial complexity of the classifier and to improve its efficiency, by the deletion of irrelevant instances in the training set, and its precision, by removing noisy instances.

Another interesting proposal is the use of weighting schemes for adjusting the distance function of the k-NN. These schemes can be applied both to the

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 176–187, 2012.

instances [3] and the features [14] of the training set. A proper set of weights for adjusting the distance function can help to train the classifier to the specific domain of the problem considered, enhancing its generalization capabilities.

A great number of the approaches proposed in recent years for improving data mining processes are related to evolutionary computation [9]. Given that the processes of performing instance selection and obtaining proper weights can be defined as search problems, evolutionary algorithms can be applied to tackle them, with promising results [4,13].

Recently, the joint application of several preprocessing techniques over a single classifier has been considered through the use of co-evolutionary algorithms [5]. The field of cooperative co-evolution [11] offers a useful framework in which several optimization techniques can be applied simultaneously, obtaining better results than those expected by using the same techniques in isolation.

In this work we present a co-evolutionary model for instance selection and instance and feature weighting, applied to the k-NN classifier (CIW-NN). This model is composed by 3 populations, where each one is focused on a specific task for improving a 1-NN classifier (instance selection, feature weighting and instance weighting). After its description, we present a full experimental study where the improvements of the model over the preprocessing techniques applied in isolation is shown. These improvements are contrasted by using nonparametric statistical tests [7], which are highly recommended for analyzing the results obtained in data mining experiments such as this one.

The rest of the work is organized as follows: Section 2 presents some preliminary concepts about the techniques used in this work. Section 3 describes the proposed model. Section 4 presents the experimental study performed for testing the behavior of CIW-NN when compared with several non-co-evolutionary techniques. Finally, Section 5 shows the conclusions arrived at.

2 Background

This section surveys some necessary preliminary concepts for describing CIW-NN. Section 2.1 presents co-evolution and some of its most interesting characteristics. Section 2.2 describes the use of instance selection in classification. Finally, Section 2.3 shows how the weighting schemes can be used for improving the precision of the classifiers.

2.1 Co-evolution

Co-evolution is the area of evolutionary computation related to techniques able to manage several different populations simultaneously. Its application consists of splitting the domain of the problem using a *divide and conquer* strategy where each population is focused on tackling a single part of the problem.

Within this field, cooperative co-evolution [11] defines how the different population can cooperate. In general, this is met by using global fitness functions which require an individual of each population for being evaluated. This allows

to benefit those individuals who behave well in cooperation with the rest of populations, in contrast with the classical fitness functions which only considers the quality of individuals in isolation.

Thus, the main motivation for using cooperative co-evolution lies in its decomposition capabilities, which can be used under several assumptions to break the *No Free Lunch* barrier present in most optimization problems [15].

2.2 Instance Selection

The main goal of instance selection [10,8] is to isolate the smallest set of instances which enable a data mining algorithm to predict the class of a query instance with the same quality as the initial data set. By minimizing the data set size, space complexity and computational cost of the subsequent data mining algorithms are reduced, improving their generalization capabilities.

It can be defined as follows: Let X be an instance where $X = (x_1, x_2, \cdots, x_M, x_c)$, with X belonging to a class c, given by X_c, and an M-dimensional space in which x_i is the value of the i-th feature of the sample X. Then, let us assume that there is a training set TR composed by N instances, and a test set TS composed by T instances. Let $RS \subseteq TR$ be the subset of selected samples that result from the execution of an instance selection algorithm. Then, each new instance T from TS can be classified by from a data mining algorithm acting over the instances of RS.

2.3 Weighting Schemes

The use of weighting schemes is another interesting enhancement for the classifiers' behavior. Although there are many different approaches for this, in this work we will focus our interest in using the weights for modifying the distance function used by the classifier.

Therefore, it is possible to define weights associated both to the features (that is, real values to weight the importance of each feature in the computation of the similarity between two instances) and to the instances (that is, real values to modify the effective distance between two instances with respect to some related properties, such as, for example, its class attribute). Both schemes have been widely studied in the past [14,3].

The final goal of the inclusion of these schemes is to improve as further as possible the precision of the classifier. Hence, most of these methods are applied through an optimization process using the original training set as reference.

3 Proposed Model

In this section we present the CIW-NN co-evolutionary model. Section 3.1 describes the different subcomponents of the model. Section 3.2 shows the fitness function designed. Finally, Section 3.3 describes the general co-evolutionary model.

3.1 CIW-NN Subcomponents

CIW-NN is based on the simultaneous search of the best possible subset of training instances, and the best possible weighting schemes for instances and features. To do so, three populations are defined and focused on three specific goals:

- Instance selection (IS): Search the best subset of training instances.
- Instance weighting (IW): Search the best weighting scheme for instances.
- Feature weighting (FW): Search the best weighting scheme for features.

Although the three populations perform a search task, they can be discriminated by several characteristics. Table 1 summarizes them:

Table 1. CIW-NN population's characteristics

Topic	IS population	IW population	FW population
Scope	Instances	Instances	Features
Codification	Binary	Real	Real
Granularity	Individual	Class	Individual
Epoch length	Simple	Multiple	Multiple
Objective	Acc./Red.	Accuracy	Accuracy

- **Scope:** Each population is focused on optimizing either instances or features.
- **Codification:** Depending on the concrete enhancement task performed, the individuals of each population will employ binary $(0, 1)$ or real $([0, 1])$ codification. This feature will define the kind of basic search method which the population will carry out, and also has a strong effect on the difficulty of the search task itself, due to real coded search spaces usually being wider and harder to explore.
- **Granularity:** CIW-NN uses two schemes of assignation of weights. Individual weights (one for each instance/feature) are assigned to IS and FW chromosomes, whereas Class weights, shared by instances of the same class, are assigned to IW chromosomes.
- **Epoch length:** CIW-NN defines how the evolution process of its populations will be scheduled, by assigning epochs of different length: Simple, that is, one generation per cycle of the global model, or Multiple, considering more than one generation. This way, CIW-NN equalizes the number of evaluations spent by each population.
- **Objective:** It refers to the objective which each population pursues. A population can cope with maximizing the accuracy obtained by the classifier, or to maximize simultaneously this accuracy and the reduction rate, that is, the ratio between the number of instances discarded and the ones that composed the original training set.

The IS population performs the search using the CHC algorithm [6], considering the configuration shown in [4], where it is highlighted as a proficient method for

this task. For improving its reduction capabilities their binary chromosomes are initialized with a certain bias, where only *prob*1 instances are selected (initialized to 1). Moreover, we have modified the original HUX crossover operator, so it only maintains *prob*0*to*1 instances selected after its application.

In the IW and FW populations the search is guided by a real coded steady-state genetic algorithm. A crossover operator with multiple descendants has been selected due to its good convergence capabilities [12]. Among all the models suggested in that study, the best results have been obtained with the operator 2BLX0.3-4BLX0.5-2BLX0.7, based on the operator BLX-α. Figure 1 depicts its application, performing 4 crossing operations with different values of the α parameter, and selecting the best offspring found. The mutation operator selected is the non-uniform one, following the recommendations of [12].

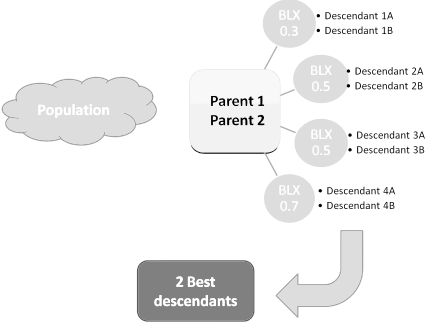


Fig. 1. Crossover operator with multiple descendants

3.2 Fitness Function

The CIW-NN fitness function is composed by two different components:

- **Accuracy:** Precision of the baseline classifier (1-NN) over the training set (using leave-one-out with the configuration of instances and weights which is evaluated).
- **Reduction:** Reduction rate of the subset of instances evaluated, over the full training set.

When performing an evaluation of the fitness function, it is required to use a chromosome from each population. If we define H as a IS population chromosome, I as a IW population chromosome, and J as a FW population chromosome, the fitness value assigned to each one is the following

$$\begin{aligned} Fitness(H) &= \alpha \cdot Ac(H,I,J) + (1-\alpha) \cdot Red(H) \\ Fitness(I) &= Ac(H,I,J) \\ Fitness(J) &= Ac(H,I,J) \end{aligned} \tag{1}$$

where $Ac(H, I, J)$ is the accuracy estimated by the classifier, $Red(H)$ is the reduction rate obtained and α is a real value $[0, 1]$ used for weighting both objectives (we have set this value to $\alpha = 0.5$, following the recommendations given in [4]).

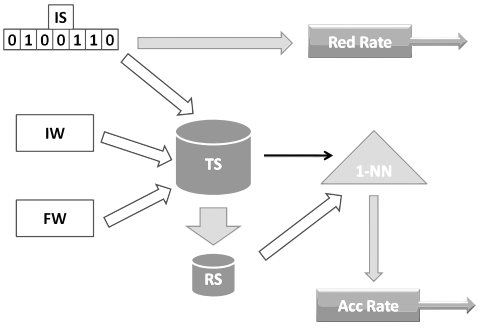


Fig. 2. Fitness function evaluation

Figure 2 shows an scheme of a fitness function evaluation. Accuracy is estimated by preprocessing the training set selecting the instances indicated by the chromosome of the IS population H, and weights defined by the IW and FW chromosomes I and J are attached to obtain the resulting set RS. This set is used as reference for the 1-NN classifier, whose accuracy $Ac(H, I, J)$ is estimated by classifying the original training set.

The similarity function used by the 1-NN classifier used to estimate the accuracy is the euclidean one. CIW-NN defines a modified version of it

$$D(x, y) = IW_{c(y)} * \sum_{i=0}^{M} FW_i \cdot \sqrt{(x_i - y_i)^2} \tag{2}$$

where x is the instance to classify, y a instance from the resulting set TS, $IW_{c(y)}$ denotes the weight assigned to the class attribute of the instance y, and FW_i denotes the weight assigned to the feature i.

3.3 Co-evolutionary Model

CIW-NN merges all the components described in the former sections in a single framework. The three populations evolves in a cycle, consuming an epoch (a fixed number of generations/evaluations) each one before passing the turn to the next population.

Figure 3 depicts the co-evolutionary scheme: The cycle is started by the IS population, performing a single generation (simple epoch). Then, the IW population performs a fixed number of generations (multiple epoch). Afterwards, the FW population performs another multiple epoch, finishing the cycle.

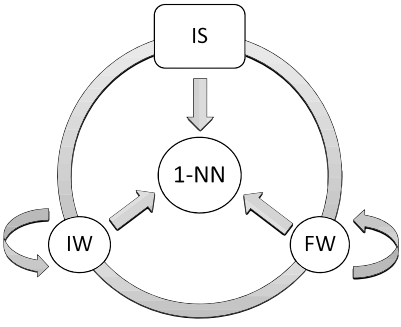


Fig. 3. CIW-NN populations scheme

```
Initialize IS, IW and FW populations;
Select the best individuals of each population;
While the limit of evaluations is not met:
     Perform an IS epoch;
     Perform an IW epoch;
     Perform an FW epoch;
     Update the best individuals found;
End
Return the best individuals found;
```

Fig. 4. CIW-NN general co-evolutionary scheme

Figure 4 summarizes the general scheme of the model. At the end of each cycle, the best individuals of each population are selected (the very first individuals - line 2 of Figure 4 - are selected according to their individual fitness). Their task will be to complement the evaluations of the new individuals generated by the search process. In this way, when a new individual must be evaluated, the best individuals selected at the two other populations are gathered, obtaining then the 3 chromosomes required by the fitness function.

This is an optimal configuration for modeling the cooperation between populations. The joint evaluation of each individual with the best individuals of the other populations allows to guide the search to more promising areas of the search space, which represent the most desirable properties of each enhancement technique. The use of the epoch model and the common fitness function allows to control how the search progresses in each component, preventing premature convergence and/or a faster convergence process of a given population to the detriment of the rest (which may lead to optimal solutions from the single point of view, but suboptimal in the cooperative sense).

Table 2. Data sets considered in the experimental study

Data set	#In.	#Ft.	#Cl.	Data set	#In.	#Ft.	#Cl.
Australian	690	14	2	Monk-2	432	6	2
Balance	625	4	3	Movement	360	90	15
Bands	539	19	2	New Thyroid	215	5	3
Breast	286	9	2	Pima	768	8	2
Bupa	345	6	2	Saheart	462	9	2
Car	1728	6	4	Sonar	208	60	2
Cleveland	303	13	5	Spectfheart	267	44	2
Contraceptive	1473	9	3	Tae	151	5	3
Dermatology	366	34	6	Tic-tac-toe	958	9	2
German	1000	20	2	Vehicle	846	18	4
Glass	214	9	7	Vowel	990	13	11
Hayes-roth	160	4	3	Wine	178	13	3
Housevotes	435	16	2	Wisconsin	699	9	2
Iris	150	4	3	Yeast	1484	8	10
Lymphography	148	18	4	Zoo	101	16	7

4 Experimental Study

In this section, we describe the experimental study performed to characterize the behavior of CIW-NN in supervised classification problems. Section 4.1 describes the data sets used. Section 4.2 enumerates the algorithms selected for the comparison and describes their parameters. Section 4.3 presents and analyze the results obtained. Finally, Section 4.4 shows the statistical study performed for contrasting the results of the experiment.

4.1 Data Sets

We have selected 30 supervised classification data sets for this study. They have been taken from the *UCI Machine Learning Repository*[1] and *KEEL-dataset Repository*[2]. Table 2 shows their main characteristics: Name, number of instances **#In.** (examples) , number of features **#Ft.** and number of classes **#Cl.**.

Every data set has been partitioned using a 10-folds cross validation procedure. Moreover, the attribute values have been normalized into the interval $[0, 1]$. This will help in equalizing the influence of every attribute with respect to the distance measure selected for the classifiers.

4.2 Algorithms and Parameters

In addition to CIW-NN, in this study we have used as comparison algorithms the three baseline methods of the populations of the co-evolutionary model:

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html
[2] http://www.keel.es/datasets.php

Table 3. Parameters of the methods

Method	Parameters
CIW-NN	α: 0.5, *prob*0*to*1: 0.25, *prob*1: 0.25, Epoch length: 40 evaluations
	Mutation probability: 0.05 per chromosome
CHC-IS	α: 0.5, *prob*0*to*1: 0.25, *prob*1: 0.25
SSGA-FW	Mutation probability: 0.05 per chromosome
SSGA-IW	Mutation probability: 0.05 per chromosome
Common parameters	Crossover operator (real): 2BLX0.3-4BLX0.5-2BLX0.7,
	Crossover operator (binary): Modified HUX
	Evaluations: 10000, Population size: 50, Base classifier: 1-NN

The CHC algorithm for IS (CHC-IS), a Steady-State Genetic Algorithm with multiple descendants for feature weighting (SSGA-FW) and a Steady-State Genetic Algorithm with multiple descendants for instance weighting (SSGA-IW). Moreover, we have included the 1-NN rule as a basic classifier for reference.

All these methods have been coded in Java, using the KEEL Software [1,2][3]. In the experimental study, we have applied a 5x10-folds cross validation procedure for evaluating their behavior. Table 3 shows the parameters considered.

4.3 Results Obtained

In the experimental study we have considered as performance measures the accuracy in test phase (accuracy when classifying new examples unseen by the classifier at the training phase) and the reduction rate obtained over the instances of TR, for those methods which are able to perform it (CIW-NN and CHC-IS).

Table 4 shows the results obtained. For each data set, the table shows the average value obtained in each performance measure. Moreover, the best result obtained in each data set is highlighted in **bold**.

Using the results of the table, we can get the following conclusions:

- The proposed approach, CIW-NN, obtains the best average accuracy. Furthermore, it outperforms all the comparison methods in 18 out of 30 problems considered.
- All the methods selected in the study greatly improves the accuracy of the 1-NN classifier.
- Both CIW-NN and CHC-IS are able to reduce the size of the training sets to less of the 10% of its original size, without harming the accuracy of the classifier.

These results supports the capabilities of instance selection and the weighting techniques for improving the performance of the 1-NN classifier. In the case

[3] http://www.keel.es

Table 4. Results obtained

Performance	Accuracy (%)					Reduction (%)	
Data set	CIW-NN	CHC-IS	SSGA-FW	SSGA-IW	1-NN	CIW-NN	CHC-IS
Australian	**81.74**	81.45	81.01	80.87	81.45	93.66	97.67
Balance	**85.75**	79.04	73.76	80.33	79.04	94.24	96.62
Bands	**75.52**	74.04	72.75	72.92	74.04	95.49	97.28
Breast	**70.62**	66.04	63.06	69.98	65.35	97.86	97.71
Bupa	60.95	62.51	**62.91**	62.29	61.08	95.36	96.55
Car	**95.89**	85.65	94.91	86.34	85.65	83.78	95.87
Cleveland	56.43	53.14	52.48	**56.45**	53.14	97.14	98.13
Contraceptive	**45.22**	42.63	44.06	44.61	42.77	84.36	97.04
Dermatology	**96.72**	95.35	96.45	94.26	95.35	96.02	96.45
German	**72.10**	70.50	69.50	71.90	70.50	89.13	97.99
Glass	**75.72**	74.50	72.36	69.35	73.61	93.25	93.51
Hayes-roth	72.15	71.01	69.96	**73.03**	35.70	91.92	92.34
Housevotes	**94.93**	91.24	93.78	91.23	91.24	97.80	98.24
Iris	93.33	93.33	**94.00**	**94.00**	93.33	96.37	95.93
Lymphography	**79.30**	73.87	76.54	77.34	73.87	94.23	94.67
Monk-2	**100.00**	95.32	**100.00**	75.09	77.91	93.29	95.40
Movement	83.06	86.39	86.67	**88.06**	81.94	74.69	88.09
New Thyroid	95.82	**97.23**	96.28	95.84	**97.23**	96.95	97.62
Pima	**71.24**	70.33	70.71	70.59	70.33	92.09	97.09
Saheart	**65.37**	64.49	64.06	64.28	64.49	96.34	97.88
Sonar	**87.00**	85.55	85.07	86.02	85.55	91.67	93.11
Spectfheart	77.92	69.70	74.63	**78.68**	69.70	98.17	97.96
Tae	65.71	65.04	**68.38**	63.04	40.50	93.82	94.41
Tic-tac-toe	87.37	82.07	**91.33**	73.07	73.07	88.67	95.62
Vehicle	**71.28**	70.10	71.16	66.55	70.10	90.28	94.48
Vowel	98.28	**99.39**	99.29	98.38	**99.39**	74.97	84.01
Wine	97.16	95.52	96.63	**97.75**	95.52	96.88	96.69
Wisconsin	96.00	95.57	95.57	**96.42**	95.57	94.74	99.21
Yeast	**52.76**	52.23	50.81	52.63	50.47	83.49	97.19
Zoo	**97.50**	96.83	96.83	95.58	92.81	89.99	89.34
Average	**80.09**	78.00	78.83	77.56	74.69	91.89	95.47

of the co-evolutionary model, this improvement is considerable, since it offers simultaneously the best results on accuracy and very high reduction rates over the training sets (which means a great improvement in the efficiency of the classifier, in terms of both storage requirements and running time).

4.4 Statistical Study

Nonparametric statistical tests for multiple comparisons may be used to contrast the experimental results achieved. Their use in data mining is specially

Table 5. Ranks of the Friedman test and p-values of the post-hoc methods

Control method: CIW-NN (Rank: 1.850)				
Method	Rank	Holm	Hochberg	Finner
CHC-IS	3.267	0.00156	0.00156	0.00104
SSGA-FW	3.117	0.00384	0.00384	0.00256
SSGA-IW	2.967	0.00623	0.00623	0.00623
1-NN	3.800	0.00001	0.00001	0.00001

recommended in those cases in which it is necessary to contrast the results of a new proposal with several comparison methods [7].

In this study, we will use the Friedman test for detecting significant differences between accuracy results. Holm, Hochberg and Finner procedures will be used as *post-hoc* tests for characterizing the differences found.[4]

After the application of the Friedman test, significant differences among the algorithms ($p = 0.00006$) are found. Hence, CIW-NN is selected as the control method (the one with the lowest rank) for the post-hoc procedures.

Table 5 summarizes the results obtained. CIW-NN improves statistically the results of the comparison methods at a $\alpha = 0.01$ level of significance (the three *post-hoc* methods obtain p-values lower than 0.01 in every case). Hence, the study contrast that the improvement of CIW-NN over CHC-IS, SSGA-FW, SSGA-IW and 1-NN is significant.

5 Conclusions and Future Work

In this work it is proposed a new approach hybridizing several data preprocessing and adjusting methods for the k-NN classifier, within a co-evolutionary framework. The experimental study performed supports the use of co-evolution as a practical tool for improving the results of the selected techniques.

Several ideas arise as future work, including the comparison of the model with a set of representative data reduction and weighting approaches of the state of the art, and the evaluation of the performance of CIW-NN in large classification domains. Moreover, the efficacy of the method could be further improved if more accurate fitness functions are developed.

Acknowledgements. This work was supported by Projects TIN2011-28488 and TIC-2010-6858. J. Derrac holds a FPU scholarship from Spanish Ministry of Education. I. Triguero holds a scholarship from the University of Granada.

[4] More information can be found at the SCI2S thematic website on *Statistical Inference in Computational Intelligence and Data Mining* http://sci2s.ugr.es/sicidm/

References

1. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M., Ventura, S., Garrell, J., Otero, J., Romero, C., Bacardit, J., Rivas, V., Fernández, J., Herrera, F.: KEEL: A software tool to assess evolutionary algorithms to data mining problems. Soft Computing 13, 307–318 (2009)
2. Alcalá-Fdez, J., Fernández, 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 17(2-3), 255–287 (2011)
3. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning. Artificial Intelligence Review 11, 11–73 (1997)
4. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. IEEE Transactions on Evolutionary Computation 7(6), 561–575 (2003)
5. Derrac, J., García, S., Herrera, F.: IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule. Pattern Recognition 43(6), 2082–2105 (2010)
6. Eshelman, L.J.: The CHC adaptative search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: Rawlins, G.J.E. (ed.) Foundations of Genetic Algorithms, pp. 265–283. Morgan Kaufmann, San Mateo (1991)
7. 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)
8. García, S., Derrac, J., Cano, J.R., Herrera, F.: Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(3), 417–435 (2012)
9. Ghosh, A., Jain, L.C. (eds.): Evolutionary Computation in Data Mining. Springer, Heidelberg (2005)
10. Liu, H., Motoda, H. (eds.): Instance Selection and Construction for Data Mining, ser. The Springer International Series in Engineering and Computer Science. Springer, Heidelberg (2001)
11. Potter, M.A., Jong, K.A.D.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evolutionary Computation 8(1), 1–29 (2000)
12. Sánchez, A.M., Lozano, M., Villar, P., Herrera, F.: Hybrid crossover operators with multiple descendents for real-coded genetic algorithms: Combining neighborhood-based crossover operators. International Journal on Intelligent Systems 24(5), 540–567 (2009)
13. Triguero, I., García, S., Herrera, F.: IPADE: Iterative Prototype Adjustment for Nearest Neighbor Classification. IEEE Transactions on Neural Networks 21(12), 1984–1990 (2010)
14. Wettschereck, D., Aha, D.W., Mohri, T.: A review and empirical evaluation of feature weigthing methods for a class of lazy learning algorithms. Artificial Intelligence Review 11, 273–314 (1997)
15. Wolpert, D.H., Macready, W.G.: Coevolutionary free lunches. IEEE Transactions on Evolutionary Computation 9(6), 721–735 (2005)
16. Wu, X., Kumar, V. (eds.): The Top Ten Algorithms in Data Mining. Data Mining and Knowledge Discovery. Chapman & Hall,CRC (2009)

Improving Multi-label Classifiers via Label Reduction with Association Rules

Francisco Charte[1], Antonio Rivera[1], María José del Jesus[1], and Francisco Herrera[2]

[1] Dep. of Computer Science, University of Jaén, Jaén, Spain
[2] Dep. of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain
{fcharte,arivera,mjjesus}@ujaen.es, herrera@decsai.ugr.es
http://simidat.ujaen.es, http://sci2s.ugr.es

Abstract. Multi-label classification is a generalization of well known problems, such as binary or multi-class classification, in a way that each processed instance is associated not with a class (*label*) but with a subset of these. In recent years different techniques have appeared which, through the transformation of the data or the adaptation of classic algorithms, aim to provide a solution to this relatively recent type of classification problem.

This paper presents a new transformation technique for multi-label classification based on the use of association rules aimed at the reduction of the label space to deal with this problem.

Keywords: Multi-label Classification, Data Transformation, Dimensionality Reduction, Association Rules.

1 Introduction

The increasing volume of documents of all kinds on the web, especially texts and images, has originated the demand to properly classify them into non mutually exclusive categories. The availability of large databases in areas such as genetics has generated the need to analyze them in order to obtain useful knowledge, as well. These and other similar circumstances have made the interest in solving this problem, called multi-label classification [8], grow in recent years.

While this is a technique originated toward the field of document categorization [17], it is also applicable to other tasks of similar interest such as map labeling [6], medical diagnosis [5] or Bioinformatics [4].

To address this new problem it has been necessary to develop new classification methods, and also to define specific measures for analyzing the characteristics of multi-label datasets and evaluating the outputs of the classifiers. In recent years, much research has addressed the multi-label classification mainly through two ways [2]: 1) reducing the problem to a set of binary or multi-class classifiers via data preprocessing techniques and 2) through adaptation of existing algorithms to provide them with the capacity to deal simultaneously with multiple labels/classes.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 188–199, 2012.

However, few researchers have addressed the problem of high dimensionality in the label space, very common in multi-label datasets, and its influence on the results and the performance of the classifiers. Some multi-label datasets have hundreds or even thousands of different labels and, sometimes, their number exceeds the amount of features and often also the number of instances. There is, therefore, a problem of high dimensionality in a new space: the label space.

In this paper a new transformation method based on association rules, *Label Reduction with Association Rules* (LRwAR), is presented. An association rule mining algorithm is applied over the label space, obtaining a set of association rules. The ones that reach a minimum confidence level are used to preprocess the dataset, reducing label cardinality as well as the total number of labels in the dataset. This subset of rules is applied, after having executed the multi-label classifier, in order to retrieve the relevant labels in a postprocessing phase.

The rest of this paper is organized as follows. Section 2 presents the fundamentals of multi-label classification and the different approaches that have been addressed in the literature. Section 3 describes our proposal, aimed at improving both performance and results of multi-label classifiers. It also briefly introduces the concept of association rules and association rule mining. The experimental study performed and the results achieved are described in Section 4. Finally, Section 5 summarizes our conclusions.

2 Multi-label Classification

In Machine Learning, and particularly in supervised learning, one of the most important applications is classification. Using a set of labeled samples (dataset) the process starts by obtaining a model that is capable of labeling new samples not seen during the learning phase. Traditionally, classifiers are designed with datasets in which each sample is associated with one and only one class or label, which is the target value to obtain once the model has been built. L being the set of labels applied to the instances of a dataset, Xi all samples belonging to a particular class and l_a and l_b the indexes of two labels, the following premises must be met:

$$L = \{l_1, l_2, ..., l_k\}, |L| = k > 1. \tag{1}$$

$$X_{l_a} \cap X_{l_b} = \oslash, \forall l_a, l_b, a \neq b. \tag{2}$$

The premise in Equation 1 indicates that the set L must have at least two classes, since otherwise all samples would be associated with the same class. Equation 2 states that instances regarding their class are disjoint subsets or, put another way, that each sample corresponds to a single class.

When $|L| = 2$ then the classification is binary and the classes are usually identified as `true` or `false`. A clear example is found in the classifiers used to process the electronic mailing separated into two categories: `spam` and `non spam`. $|L| > 2$ specifies a multi-class classifier. Whereas a binary classifier would provide a positive or negative output, a multi-class classifier returns a value that, after the appropriate interpretation, will determine the class. In any case each instance belongs to only one of these classes.

Addressing a multi-label classification problem [8] Equation 2, which states that the instances subsets must be disjointed with respect to the class they belong to, is not satisfied, i.e. a sample can belong to more than one class. As indicated in Equation 3 the classifier, however, once trained will facilitate a set Y, subset of L, with the labels associated with each test instance. The premise in Equation 2 leads to the expression in Equation 4, telling us that subsets of instances are not necessarily disjointed with respect to their class.

$$Y = f(x_i), Y \subseteq L. \tag{3}$$

$$\neg\forall l_a, l_b, X_{l_a} \cap X_{l_b} = \oslash. \tag{4}$$

While in traditional classification the goal is to associate each sample to a class between $|L|$ possible classes, so that the range of possible output values is limited by the number of existing classes, in a multi-label classifier there are $2^{|L|}$ different possible values as output: it could be any combination of labels in L. A multi-label classifier, as stated in [8], generates its prediction in one of two ways: with a binary partition of the label set or with a label ranking.

The traditional classification algorithms based on trees, neural networks, support vector machines and instances, are designed to provide a single value as output: the class which the processed sample belongs to. These algorithms cannot be used directly, as such, to tackle a problem of multi-label classification. The literature [8] proposes two different ways to deal with this problem:

- To transform the dataset, making it possible to use the known classification algorithms, such as training a binary classifier for each label.
- To adapt a traditional classification algorithm, adding the ability to deal with the fact that each sample can be associated with multiple labels.

Each approach brings benefits but also has disadvantages that it is necessary to know in order to choose the best option.

2.1 The Data Transformation Approach

While many data transformation based methods have been proposed (a complete taxonomy can be found in [2]) Binary Relevance (BR) and Label Powerset (LP) are the two most important. These methods are algorithm independent and also known as *problem transformation methods*.

Introduced in [19] with the name *ensemble of binary classifiers*, the BR method divides the multi-label dataset into multiple binary datasets. An independent binary classifier will be trained for each dataset. There will be as many binary classifiers as different labels the original dataset had. Test samples will go through each binary classifier and finally, by the union of all the binary predictions, the multi-label prediction will be obtained. It is a simple method that allows us to use any underlying binary classification algorithm. Its main drawback is that it dismisses the relationship between labels, information potentially useful for improving the results in classification. It also implies a linear increase in execution time by the total number of labels in the dataset.

The LP method was introduced in [14] under the name *MODEL-n*. The final n highlights the fact that each distinct combination of labels in the data becomes a *new* class. Thus the original multi-label dataset becomes a classical one in which each data instance is associated with only one class, allowing the use of any multi-class classification algorithm. Unlike BR, the LP method takes into account the relationship between the labels by generating a new class for each different combination. The main problem with this method is that the number of combinations of labels is $2^{|L|}$, so the amount of classes could become intractable.

2.2 The Method Adaptation Approach

The transformation methods described above allow us to addressing the problem of multi-label classification using algorithms that are not designed for the specificities of the task. Faced with this choice, the focus of the algorithm adaptation approach aims to modify existing algorithms so that they can deal with multi-label samples, without requiring any preprocessing. In recent years the number of proposals published in this regard has increased strikingly. So here we just list the more remarkable ones.

In [4] the authors modified the C4.5 algorithm in order to classify genes according to their function, which can be multiple. They made two changes: each leaf of the tree stores not a class but a set of them, and the original entropy measure is adapted to take into consideration the fact that the samples are multi-label. Another tree-based algorithm is proposed in [7]. This begins with an ADT (Alternate Decision Tree) and incorporates an internal decomposition of the multi-label samples by OVA (One-vs-All) technique.

There are several adaptations of instance based algorithms. The most notable are ML-kNN[15] and IBLR-ML[20], the latter being a variation of the first. ML-kNN produces a ranking of labels from the closest neighbours of the instance to classify and basically operates as a binary classifier for each label, regardless of the relationships between them. IBLR-ML addresses this deficiency by considering the labels of nearest neighbours as additional input features to the classifier.

The first adaptation of a neural network to multi-label classification was BP-MLL[16], a perceptron with back-propagation learning which introduces a modified error function that takes into account the multi-label nature of the samples. Another proposal in this field is the ML-RBF[18] algorithm for designing multi-label RBFNs. The number of neurons in the hidden layer is calculated according to the number of labels in the dataset, obtaining their centers by the K-means algorithm. The training, which adjusts the weights of the connections between neurons, is performed with the SVD (Singular Value Decomposition) method and an adapted error function.

There are also proposals based on Support Vector Machine, such Rank-SVM[3], classifier chains[13] and ensembles of classifiers[9] and even those based on ant colonies, like MuLAM[1]. The number of publications related to the adaptation of algorithms for multi-label classification is constantly growing.

2.3 New Measures and Evaluation Metrics

The peculiarities of the problem faced require the use of new measures, firstly, to characterize the multi-label datasets, and secondly to facilitate the evaluation of the new algorithms.

In the first group there is *label cardinality* (Equation 5), defined as the average number of labels per sample in the dataset D, and *label density* (Equation 6), defined as label cardinality divided by $|L|$ (the total number of labels), providing a measure independent of the absolute number of labels in the dataset. These metrics offer information about multi-label datasets useful for fine-tuning the operation of the algorithms, as discussed later.

$$Card(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} |Y_i|. \tag{5}$$

$$Dens(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i|}{|L|}. \tag{6}$$

In terms of measures that assess the quality of predictions, they can be grouped 1) by operating on a bipartition of the labels or on a ranking of these, and 2) according to the calculation method: averaging by instance (example based) or label (label based). There are more than a dozen different measures that are exposed in detail in [8]. One of the most widely used, and taken as reference in this work, is *Hamming Loss* (Equation 7), where Y_i is the set of predicted labels, Z_i the set of real labels and the operator Δ represents the symmetric difference.

$$HammingLoss = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \Delta Z_i|}{|L|}. \tag{7}$$

3 LRwAR: Label Reduction with Association Rules

This section is devoted to describing LRwAR. Section 3.1 describes the problem of high dimensionality in the label space in multi-label datasets. Section 3.2 shows our approach to tackling this problem with a novel method of data transformation based on association rules.

3.1 The Problem of Dimensionality in the Label Space

Label cardinality (previously defined) is a factor that plays an important role in the performance of classifiers, affecting both their efficiency and effectiveness. By using transformation techniques like those described in Section 2, with the BR method a higher cardinality implies more binary classifiers must be used, so the computational complexity increases. The number of label combinations also increases, which affects the LP method and generally results in a reduction in

classification accuracy. Therefore it seems logical to think that if there were a way to reduce the cardinality when there is a problem of high dimensionality in the label space, the classifier performance could improve globally.

Algorithms adapted to the multi-label problem are also affected by the problem of high dimensionality in the label space. It is important to keep in mind that many of these algorithms are based on an internal data transformation similar to those described above, of which the most usual is based on training multiple binary classifiers and combining their results. Also, the predictive models obtained are much more complex with a larger number of labels and high cardinality datasets. With the reduction of cardinality these models will be simpler, more efficient and more accurate.

This paper proposes a methodology based on the transformation of data for multi-label classification which uses association rules. Association rules[12] describe co-occurrences between elements that provide knowledge about the underlying system to a set of data and can be interpreted as implications, so that the presence of certain elements implies the occurrence of others. The process of extracting association rules from a database or set of transactions is known as association rule mining, and there are many algorithms for doing this. In [12] there is a review of the most common ones, many of them based on the best known: the Apriori algorithm. As stated in [11], FP-Growth is a better association rule mining algorithm for working with large transaction databases.

Our interest in the association rules is related to their use as a tool for hiding to the multi-label classifier labels that can be inferred, with a certain confidence level, from the appearance of other labels. For this we used the so-called *support-confidence framework* defined in the literature.

3.2 LRwAR: A Method for Reducing Dimensionality in the Label Space

Label Reduction with Association Rules (LRwAR) is an original method that, through the application of preprocessing and postprocessing stages and based on the extraction of association rules, can reduce the cardinality of multi-label datasets. This reduction improves the performance and results obtained by the underlying classifiers. Operating exclusively in that label space (ignoring all other attributes), we assume that each label is an *item* and that the labels assigned to each instance of the dataset form a *transaction*. The objective is to obtain, with the FP-Growth algorithm, rules that allow the inference of certain labels from the presence of others with a certain level of confidence. The inferred labels can be hidden from the multi-label classifier. Hidden labels are added to each instance after the classifier's prediction in a postprocessing phase, simply by applying the previously-obtained association rules.

The number of rules obtained from a multi-label dataset is a parameter that depends not on the total number of different labels it contains, but on label cardinality. If that measure is small, close to 1, transactions are mostly formed by 1 or 2 items and it will be difficult to obtain valid rules.

The application of each obtained rule involves removing one or more labels from the dataset. It should be noted that apart from reducing the total number of labels in the dataset the most affected parameter, significantly reduced, will be label cardinality. This will help to reduce both execution time and complexity of models generated and, in theory, should also improve classification results.

Method Description. The proposed transformation method, including preprocessing and postprocessing stages, is described in the following pseudocode:

```
1. X = Dataset to process
2. T =   # Label set per instance (transactions)
3. For each instance Xi in X
4.   Li = labels of Xi
5.   T = T ∪ Li
6. R = FPGrowth(T) # Rules set ordered by confidence
7. MR = SubC(R)    # Get better rules in R
8. C = LabelsInConsequent(MR)
9. X = X - C   # Hide labels in C from original dataset
10. DTra = TrainingPartition(X)
11. DTst = TestPartition(X)
12. Clas = ObtainClassifier(DTra)
13. For each instance Xi in DTst
14.   Pi = Clas(Xi) # Obtain classifier's prediction
15.   Pi = Pi ∪ ApplyRules(MR) # Label inference
16. Evaluate(P)
```

The FPGrowth algorithm (line 6) returns a set of association rules ordered by a certain quality measure, in our case confidence. From this set the method obtains a subset (line 7) with the best rules, those that achieve a minimum value in this measure.

Once the classifier `Clas` is obtained, and when the test partition processing starts, the postprocessing step comes into the picture. This, as shown in line 15, consists of the application to the prediction of the classifier for each test sample the rules used when the training was realized, inferring the labels necessary to add. Finally, given `P` the total set of predictions for the test samples, the calculation of measures of quality assessment is carried out.

Method Implementation. The method described above has been implemented in Java and integrated with the MULAN[10] software, so that the process of partitioning the dataset, classifier training and test outcome rests entirely on MULAN.

The preprocessing stage creates a new multi-label dataset, having deleted the labels that can be inferred from chosen rules. That reduced dataset is given as input to MULAN. In order to integrate the postprocessing step a specialized class has been derived from MULAN's `Evaluator` class, adding labels inferred

from the rules after the classifier's prediction but before evaluation measures calculation.

MULAN stores for each processed sample the label's associated bipartition and also the measure associated with each of them that will serve to build the label ranking. LRwAR delivers to MULAN an updated bipartition while post-processing, in the application of association rules phase, including rule confidence as the measure for ranking generation.

4 Experimental Framework, Results and Analysis

This section describes the experimental study conducted for this paper. Datasets, algorithms and parameters are detailed in Section 4.1. Results are shown and discussed in Section 4.2.

4.1 Experimental Framework

In order to empirically test the proposed method we selected a set of MULAN [10] repository's datasets with mixed characteristics: many vs. few labels, high vs. low cardinality, etc. Table 1 indicates the name of each dataset and its main characteristics in instance: total number and number of distinct instances (those which have different values in at least one attribute), attribute: number of nominal and numeric attributes, and label spaces.

Table 1. Datasets used in experimentation and their characteristics

Dataset	Instances		Attributes		Labels		
Name	Number	Distinct	Nominal	Numeric	Total	Cardinality	Density
emotions	593	27	0	72	6	1.869	0.311
CAL500	502	502	0	68	174	26.044	0.150
Corel5k	5000	3175	599	0	374	3.522	0.009
enron	1702	753	1001	0	53	3.378	0.064
yeast	2417	198	0	103	14	4.237	0.303

The execution of the FP-Growth algorithm on the labels of each dataset, in order to extract association rules, was performed with the following parameters: support was set to 0.025 and minimum confidence at 0.5.

As for multi-label classification algorithms used for this study, we wanted to have a representation of methods based on both data transformation and method adaptation approaches. It should be noted that the transformation methods do their work after the preprocessing phase described previously, operating on a reduced dataset having already eliminated the labels inferred from the rules. The methods chosen, all implemented in MULAN, were:

– **BR-J48/LP-J48:** Performs a BR or LP transformation, respectively, and uses the classification algorithm C4.5 (called J48 in Weka/MULAN) as underlying algorithm for binary classification.

- **MLkNN:** This algorithm was proposed in [15], it is a k nearest neighborhood instance based method.
- **IBLR-ML:** An improved version of the previous method, presented in [20]. Combines instance based classification and logistic regression.
- **BP-MLL:** This algorithm was proposed in [16]. It is a back-propagation neural network based method.

Each run of the proposed method, once the set of rules to apply has been obtained and the dataset preprocessed, is repeated 5 times for each classification method with the instances of the dataset distributed randomly at the beginning of each repetition. Cross-validation is used with the usual configuration at 10 partitions, so there is a total of 5 repetitions x 10 partitions = 50 runs of the underlying algorithm for each dataset. The results are averages of those 50 runs, obtaining for every one the HammingLoss (HL) measure described in Section 2.

4.2 Results and Analysis

Table 2 shows the results obtained with transformation methods: LP and BR, whereas Table 3 shows the results from methods adapted to multi-label classification. In both tables the first column of every algorithm corresponds to the measure (HL metric) of the base method, the second ($LRwAR_1$) is the result obtained by applying all the obtained rules and the third ($LRwAR_2$) after the application of only the most confident association rule. The best results are highlighted in bold.

Table 2. Results from transformation methods

Algorithm	LP-J48			BR-J48		
Dataset	Base	$LRwAR_1$	$LRwAR_2$	Base	$LRwAR_1$	$LRwAR_2$
emotions	0.2777	0.2770	**0.2476**	**0.2474**	0.2793	0.2584
CAL500	0.1996	**0.1991**	0.1999	0.1615	**0.1607**	0.1618
Corel5k	0.0168	**0.0161**	0.0167	0.0098	**0.0097**	0.0098
enron	0.0716	0.0761	**0.0715**	**0.0508**	0.0581	0.0533
yeast	0.2780	0.2808	0.2780	**0.2454**	0.2508	0.2468

Table 3. Results from adapted methods

Algorithm	MLkNN			IBLR-ML			BP-MLL		
Dataset	Base	$LRwAR_1$	$LRwAR_2$	Base	$LRwAR_1$	$LRwAR_2$	Base	$LRwAR_1$	$LRwAR_2$
emotions	0.1951	0.2299	**0.1904**	0.1883	0.2205	**0.1872**	0.2061	0.2633	**0.1999**
CAL500	0.1388	**0.1387**	0.1389	0.2307	**0.2300**	0.2315	**0.2478**	0.2509	0.2506
Corel5k	0.0094	**0.0093**	0.0094	0.0225	**0.0219**	0.0229	0.8315	0.7735	**0.7631**
enron	**0.0524**	0.0573	0.0543	**0.0557**	0.0610	0.0573	**0.3831**	0.6424	0.5443
yeast	0.1933	0.1933	**0.1931**	0.1934	**0.0092**	0.1928	0.2258	0.2264	**0.2244**

From the analysis of these results it can be observed that the proposed method obtains 17 best results against 7 cases with no improvements. Our proposal outperforms all the base methods except for BR-J48.

If we analyze the datasets from their characteristics, Table 1, we appreciate that for datasets with the largest number of labels and higher cardinalities, such as `CAL500` (174 labels) and `Corel5k` (374 labels), the label reduction applying all the obtained rules achieves the best results. By contrast, for datasets with a low number of labels, such as `emotions` and `yeast`, our method improves the results by applying only the most confident rule.

It seems logical to conclude, therefore, that the elimination of the maximum number of labels (applying all the obtained rules) and reduction of label cardinality, has a positive influence when the cardinality and total number of labels is large, whereas the use of only one rule is more appropriate for datasets with fewer labels and a minor label cardinality.

It should be noted that these results, which maintain or even improve the evaluation measures, are obtained in a shorter run time and generate simpler models than the original, since they work with a reduced version of the datasets. Is a fact that becomes clear in Table 4, which shows the average execution times in seconds for every algorithm-dataset combination. Our proposal improves in all cases except for the Corel5k-MLkNN combination. It must be highlighted that MLkNN is a lazy sort algorithm that does not generate a prediction model, most of its run time is used to calculate distances between the input variables of the sample data space, a space that is not modified by our proposal.

Table 4. Average run time (seconds)

Algorithm	LP-J48		BR-J48		MLkNN		IBLR-ML		BP-MLL	
Dataset	Base	LRwAR	Base	LRwAR	Base	LRwAR	Base	LRwAR	Base	LRwAR
emotions	6.93	**1.87**	4.49	**1.43**	3.50	**3.27**	5.59	**4.36**	7.88	**4.49**
CAL500	71.34	**68.58**	14.97	**14.41**	4.39	**3.79**	316.38	**283.53**	277.79	**260.95**
Corel5k	5333.12	**4442.04**	448.52	**445.82**	**1244.7**	1333.03	17870.88	**16648.92**	4815.05	**4004.86**
enron	2174.22	**1859.17**	125.81	**125.11**	44.07	**43.80**	150.84	**122.91**	1553.37	**1501.86**
yeast	144.99	**114.01**	107.22	**86.15**	112.62	**101.59**	138.62	**121.6**	66.19	**56.9**

5 Conclusions and Future Work

In this work we have presented a novel transformation method designed to reduce the number of labels in a multi-label dataset, as well as the label cardinality, by the use of association rules. This approach can be used with any underlying multi-label classification algorithm, allowing classifier training in less time, resulting in simpler models and, in many cases, improving evaluation measures.

Experimentation and results (Section 4.2) lead us to conclude that it is a useful approach to reducing label cardinality of multi-label datasets. Adjusting the optimal number of association rules to apply, depending on the characteristics of the processed dataset, needs a deeper analysis. Both the definition of the proposed method and the experimentation are a first approach to dimensionality reduction in the label space, an alternative studied poorly until now.

Acknowledgments. Supported by the Spanish Ministry of Education under the F.P.U. National Program (Ref. AP2010-0068), the Spanish Ministry of Science and Technology under the Project TIN2008-06681-C06-02, FEDER founds, and the Andalusian Research Plan TIC-3928, FEDER founds.

References

1. Chan, A., Freitas, A.A.: A new ant colony algorithm for multi-label classification with applications in bioinfomatics. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 27–34 (2006)
2. de Carvalho, A., Freitas, A.: A Tutorial on Multi-label Classification Techniques. Foundations of Computational Intelligence 5, 177–195 (2009)
3. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. Neural Information Processing Systems, 681–687 (2001)
4. Clare, A.J., King, R.D.: Knowledge Discovery in Multi-Label Phenotype Data. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, p. 42. Springer, Heidelberg (2001)
5. Karalic, A., Pirnat, V.: Significance level based multiple tree classification. Informatica 15(5) (1991)
6. Zhu, B., Poon, C.K.: Efficient Approximation Algorithms for Multi-label Map Labeling. In: Proceedings of the 10th International Symposium on Algorithms and Computation, pp. 143–152 (1999)
7. Comité, F., Gilleron, R., Tommasi, M.: Learning Multi-label Alternating Decision Trees from Texts and Data. In: Perner, P., Rosenfeld, A. (eds.) MLDM 2003. LNCS, vol. 2734, pp. 35–49. Springer, Heidelberg (2003)
8. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685 (2010)
9. Tsoumakas, G., Vlahavas, I.P.: Random k -Labelsets: An Ensemble Method for Multilabel Classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 406–417. Springer, Heidelberg (2007)
10. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: MULAN: A Java Library for Multi-Label Learning. Journal of Machine Learning Research, 2411–2414 (2011)
11. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. of the 2000 ACM-SIGMOD International Conference on Management of Data, vol. 29(2), pp. 1–12 (2000)
12. Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining - a general survey and comparison. ACM SIGKDD Explorations Newsletter 2(1), 58–64 (2000)
13. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier Chains for Multi-label Classification. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS Part I, vol. 5782, pp. 254–269. Springer, Heidelberg (2009)
14. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recognition 37, 1757–1771 (2004)
15. Zhang, M., Zhou, Z.: ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition 40, 2038–2048 (2007)

16. Zhang, M., Zhou, Z.: Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. IEEE Transactions on Knowledge and Data Engineering 18, 1338–1351 (2006)
17. Zhang, M., Zhou, Z.: A k-nearest neighbor based algorithm for multi-label. In: Proceedings of the 1st IEEE International Conference on Granular Computing, pp. 718–721 (2005)
18. Zhang, M.: Ml-rbf: RBF Neural Networks for Multi-Label Learning. Neural Processing Letters 29, 61–74 (2009)
19. Godbole, S., Sarawagi, S.: Discriminative Methods for Multi-labeled Classification. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 22–30. Springer, Heidelberg (2004)
20. Cheng, W., Hüllermeier, E.: Combining Instance-Based Learning and Logistic Regression for Multilabel Classification. Machine Learning 76, 211–225 (2009)

A GA-Based Wrapper Feature Selection for Animal Breeding Data Mining*

Olgierd Unold[1], Maciej Dobrowolski[2], Henryk Maciejewski[1], Pawel Skrobanek[1], and Ewa Walkowicz[2]

[1] Institute of Computer Engineering, Control and Robotics
Wroclaw University of Technology, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
{olgierd.unold,henryk.maciejewski,pawel.skrobanek}@pwr.wroc.pl
[2] Department of Horse Breeding and Riding,
Wroclaw University of Environmental and Life Sciences
Kożuchowska 6, 51-631 Wrocaw, Poland
{maciej.dobrowolski,ewa.walkowicz}@up.wroc.pl

Abstract. Feature selection methods are used to tackle the problem of the curse of the dimensionality of data to be mined. This applies also to the area of animal breeding, in which datasets collect remarkably a large number of animal features. In this paper, we have conducted a comprehensive study of both 12 classification methods as well as 12 GA-based feature selection methods for classification of the Silesian horse data. To assess the performance of the wrappers and the classification methods over the animal dataset we used two metrics: a probability metric Area under the ROC curve (AUC), and a rank metric Root Mean Square Error (RMSE). All of the classifiers and the wrappers were taken from the Weka machine learning software. We find that most of the GA-based wrappers achieved results no worse than high-dimensional dataset. The statistical results obtained make the three classifiers: a decision tree ADT, a logistic regression Log and a bagging method Bag competitive method to be considered in the field of animal breeding data mining.

Keywords: Feature selection, Genetic algorithm, Data Mining, Breeding.

1 Introduction

The curse of the dimensionality (CofD) refers to the fact that the sample size needed to estimate a function of several variables to a given level of accuracy grows exponentially with the number of variables [1]. Another problem with the CofD, especially in data mining, is that it is one of tha main causes why classifiers over-fit the training data. A way to tackle the problem of CofD is to shrink the input dimension of the function to be estimated. Note that, in many situations, it is possible to reject the redundant information, because of

* This work was conducted as part of research project no. N516 415138 financed by the Ministry of Science and Higher Education.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 200–209, 2012.

(1) some of the variables may be correlated through linear combinations or other functional dependence with each other, and (2) some of the variables may have a variation smaller than the measurement noise and therefore may be irrelevant.

Scientists in many fields face the problem of simplifying high-dimensional data by finding low-dimensional structure in it. This also applies to the area of animal breeding, in which datasets collect very often a large number of animal features. There is few study on the use of data minig in phenotype driven animal breeding [6]. Although in [2] the performance of 6 classifiers over a horse breeding problem was investigated, the authors used genetic information (microsatellite markers) to perform a prediction of an individuals breed. To our knowledge an extensive evaluation of both feature selection as well as machine learning methods in the context of phenotype driven animal breeding data has not been conducted.

In this work, we carry an elaborate performance study of both different classification methods and different feature selection methods applied to Silesian horse dataset [10]. We investigate 12 classification methods, and for feature selection, we use and compare 12 GA-based wrapper approaches. We find that while most feature selection approaches give similar results, without impairing the results in comparison with an input, non-reduced dataset, there are three distinct classification methods outperforming all other chosen ones.

For the study, we used feature and classification methods taken from the standard Weka[1] software [11], and from the fuzzy-rough version of Weka[2] [8].

The remainder of this paper is organized as follows. Section 2 describes animal breeding data, a set of classifiers and GA-based wrapper feature selection methods mining the data, also Weka software used in this work. Section 3 shows the results obtained, and finally the conclusions are drawn in Section 4.

2 Data and Methods

2.1 Data

In our work, we used Silesian horse dataset [10]. This database consists of 18.980 records containing zoometric and breeding information of Silesian horses released in Poland in the last 50 years. The attributes depict such features of the horse as: the year of birth, the geographical location (the breeder), type of ownership (private and national), the assignment to the breeding program, origin (identification key of a father, identification key of a mother, family, race), relationship (offspring, inbreeding - connexion with the other records), sex, size, zoometric indexes, bonitation. Most of the entries in the Silesian horse database is information about the mares - 13.408 records, the remaining 5.572 records are the data for stallions.

A data mining goal was to predict the height at withers of the horse (or more precisely the height at withers of the mare) only on the basis of her parents.

[1] http://www.cs.waikato.ac.nz/ml/weka/, last accessed 13 November 2011.

[2] http://users.aber.ac.uk/rkj/book/wekafull.jar, last accessed 13 November 2011.

The height at the withers is a crucial factor that determines the usefulness of Silesian horses. After the necessary transformation of the dataset, we obtained the database divided up into 4.854 records and 69 attributes. The transformation was based on a joining all the attributes of the mother, father and offspring to one record, and then removing all data and string attributes as non-informative ones. Among these 69 attributes, 28 were nominal. The dataset contained missing values. The target feature was binary: 0 for the horse with the height less than the average of a population, and 1 for the height more than the average of the population. The dataset was balanced 2.521/2.333 records with the target feature of 0/1, respectively. Note that we are not interested in predicting the exact height of an offspring, and only the class of the height at withers of a child.

2.2 Classifiers

We conducted experiments comparing the classification accuracy and error of 12 classification methods implemented in the Weka software. The set of used classifiers can be grouped into the following categories:

- *Naive Bayes* classifiers based on the Bayesian Theorem in which it is assumed that the attributes have equal weight and are conditionally independent,
- *Support vector machines* trying to find a hypersurface in the space of possible inputs,
- *Decision trees* creating a hierarchy of nodes, each associated with a decision rule on one attribute,
- *Decision rules* generating rules, which can transformed from or in decision trees,
- *Nearest Neighbor* classifiers (known as instance-based classifiers) using the k nearest neighbors in the feature space to decide which class an object belongs to,
- *Logistic Regression* classifier based on searching for a dependence of the target variable in the form of a logistic function,
- *Bagging* models trained on bootstrap replicates of the training data are combined by voting.

For more information on implemented in the Weka software classifiers see [11].

2.3 GA-Based Wrapper Feature Selection

Feature selection (FS) methods can be put into three categories from the point of view of a methods output. One category is about ranking feature according to the same evaluation criterion (filter approach); the other is about choosing a minimum subset of features that satisfies an evaluation criterion (the wrapper approach), and the last regularizes predictor estimation by constraining the dimension of the input space (the embedded approach).

The wrapper approach produces the best results out of the FS methods [12], although this is a time-consuming method since each feature subset considered

must be evaluated with the classifier algorithm. In the wrapper method, the attribute subset selection algorithm exists as a wrapper around the data mining algorithm and outcome evaluation. The induction algorithm is used as a black box. The FS algorithm conducts a search for a proper subset using the induction algorithm itself as a part of the evaluation function. GA-based wrapper methods involve a genetic algorithm (GA) as a search method of subset features. GA is a random search method, effectively exploring large search spaces [7]. The basic idea of GA is to evolve a population of individuals (chromosemes), where individual is a possible solution to a given problem. In case of searching the appropriate subset of features, a population consists of different subsets evolved by a mutation, a crossover, and selection operations.

To perform GA-based wrapper feature selection we used a *Wrapper Subset Evaluator* toolbox included in the Weka software. This takes as a parameter the name of the classifier being used to evaluate attribute sets. To estimate the accuracy of a chosen classifier for an examined set of attributes the 5-fold cross validation is used (the Weka default parameter). The Weka *GeneticSearch* method performs a search for attribute subsets using a genetic algorithm. The evaluated population by GA consists of chromosomes, where each chromosome is a list of attribute indexes. After the initial population is chosen randomly, the algorithm evolves (using genetic operators) in such a way that the fitness of chromosomes (assessed by a classifier) increases over the generations. After reaching maximum generations, algorithms returns the chromosome with the highest fitness (in other words the subset of attributes with the highest accuracy).

2.4 Experimental Settings

For our evaluation of different classification methods we used 12 different classifiers from the Weka software: two Naive Bayes classifiers (NaiveBayes **NB** and BayesNet **BN**), one type of Support vector machine (**SMO**), one rule-based classifier (**JRip**), three types of decision trees (C4.5 **C45**, ADTree **ADT**, and Random Forest **RF**), three k-nearest neighbour algorithms (**IBk**, fuzzy nearest neighbour algorithm **FNN**, and fuzzy-rough nearest neighbour algorithm **FRNN**), one regression model (Logistic regression **Log**), and one bagging model **Bag**. All classifiers were trained using the Weka default parameters.

To investigate the impact of feature selection on the classification performance, 12 wrapper approaches were applied to the dataset. GA is used as random search method with 12 mentioned above different classifiers as induction methods.

Two metrics are used to assess the performance of wrappers and classification methods: RMSE and AUC. AUC is a probability metric, and RMSE a rank metric.

Root Mean Square Error (RMSE) is a metric corresponding to the expected value of the squared error loss or quadratic loss [3]. RMSE is a frequently used measurement of the differences between values predicted by a model or an estimator, and the values actually observed in what is being modelled or estimated.

Area under the ROC curve (AUC) measures the area under a plot of the fraction of positive examples misclassified on the x axis against the fraction of positive examples correctly classified [4]. The AUC of a classifier is equivalent to the probability that the classifier has of ranking a randomly chosen positive instance higher than a randomly chosen negative instance. AUC was proved to be a better measure than accuracy when evaluating and comparing classifiers [9].

The data mining was conducted as follows: first, feature selection was performed on a given dataset. Each GA-based wrapper method used 5-fold cross validation protocol (the Weka default parameter). Then, the optimal feature subset indicated by each wrapper was used as an input data for different classifiers (induction methods). Each induction method split random the data into training and validation parts (in proportion 2/3 and 1/3, respectively), and the entire induction process was repeated 10 times.

The results of the Shapiro-Wilk tests rejected the hypothesis concerning the origination of every tested variable from a normal distribution. The non-parametric Friedman test [5] was used to test the null hypothesis between dependent groups (in each tested group we use the same list of the wrappers or the classifiers). For AUC and RMSE levels, we compared the values between groups using Friedman analysis. Friedman test is a multisample extension of the sign test, a non-parametric randomized block analysis of variance, free from the assumptions of normal distribution and equal variances. The R[3] software package was used for statistical computing.

3 Experimental Results

The performance of the 12 GA-based wrappers applied to the dataset is shown in Table 1. As a result, we obtained 12 datasets containing different subsets of features, and additionally the input dataset without dimension reduction (WRP.no). Most of the GA-based wrappers reduced the dimension ca. by a half. The minimum number of features contains the dataset performed by WRP.IBk (13 attributes).

The mean performance of the feature selection methods is shown in Figure 1 and Figure 2. We calculated boxplots of the AUC (Figure 1) and of RMSE (Figure 2) of each of the 12 wrappers plus the dataset without reducing over all classifiers (WRP.no). All boxplots in Figure 1 were ordered by decreasing AUC mean, and in Figure 2 by increasing RMSE mean. Mean AUC ranges between 0.74 (for WRP.C45) to 0.72 (for WRP.FRNN) .

Furthermore, all methods show extreme outliers caused mainly by Ibk induction method (for 8 wrappers), and next FNN (for 4 wrappers).

The AUC levels in wrapper WRP.ADT and WRP.C45 outperformed AUC levels of WRP.FRNN (P=0.05, Friedman test).

The RMSE boxplots indicate that most of the wrappers gained similar mean RMSE, varies from 0.483 for WRP.IBk to 0.497 for WRP.FRNN. The interquartile ranges (IQRs) of AUC and RMSE boxplots demonstrate the significant

[3] http://www.r-project.org/

Table 1. Results of feature selection of GA-based wrappers (incuding dataset without dimension reduction - WRP.no)

Wrapper	Number of attributes
WRP.no	69
WRP.SMO	40
WRP.JRip	36
WRP.BN	35
WRP.C45	33
WRP.FNN	32
WRP.RF	31
WRP.Log	31
WRP.NB	29
WRP.Bag	29
WRP.FRNN	28
WRP.ADT	26
WRP.IBk	13

volatility among classifiers. As in the case of AUC, all extreme outliers are caused by two classifiers: Ibk (for 8 wrappers) and FNN (for 5 wrappers).

RMSE levels are not significantly different among wrappers, except wrapper WRP.ADT (P=0.5, Friedman test).

The mean performance of the classifier induction methods is shown in Figure 3 and Figure 4. The AUC boxplots, as well as RMSE boxplots, are no longer so aligned as in case of wrapper boxplots. The IQRs of AUC and RMSE boxplots are significant lower in comparison to wrapper boxplots indicating reduced variability among datasets. Mean AUC ranges between 0.79 (for WRP.Log) to 0.64 (for WRP.Ibk). All classifiers show moderate outliers (casued mainly by WRP.Ibk, which reduced the input dataset from 69 into 13 attributes).

The post-hoc test showed (P=0.05, Friedman test) that AUC levels in ADT and Log were significantly higher as compared with 5 other classifiers: FNN, FRNN, Ibk, Jrip, SMO, the AUC levels of Bag and BN higher than in 4 classifiers: FBB, FRNN, Ibk, SMO, the AUC levels in NB higher as compared with 3 classifiers: FNN, FRNN, Ibk, the levels of AUC in RF higher than in 2 classifiers: FNN and Ibk, and the AUC levels in C45 significant higher than in Log.

The mean RMSE ranges between 0.429 (for WRP.Log) to 0.596 (for WRP.Ibk). The RMSE levels were significantly lower in the classifiers ADT, Log, and Bag as compared with others classifiers: ADT classifier - BN, FNN, Ibk, NB, SMO; Log classifier - FNN, FRNN, Ibk, NB, and SMO; Bag classifier - BN, FNN, FRNN, Ibk, NB, and SMO (P=0.05, Friedman test). The levels of RMSE in Jrip classifier are significantly lower than in 4 other classifiers: FNN, Ibk, NB, and SMO. The RMSE levels in C45 classifier are lower as compared with 3 classifiers: FNN, Ibk, SMO. The RF classifier has the lower RMSE levels than 2 classifiers: FNN and SMO. THE RMSE levels in FRNN classifier are lower than in two classifiers: FNN, Ibk. After all, levels of RMSE in Ibk classifier are significantly lower as compared with RF. All post-hoc tests used Friedman test and P=0.05.

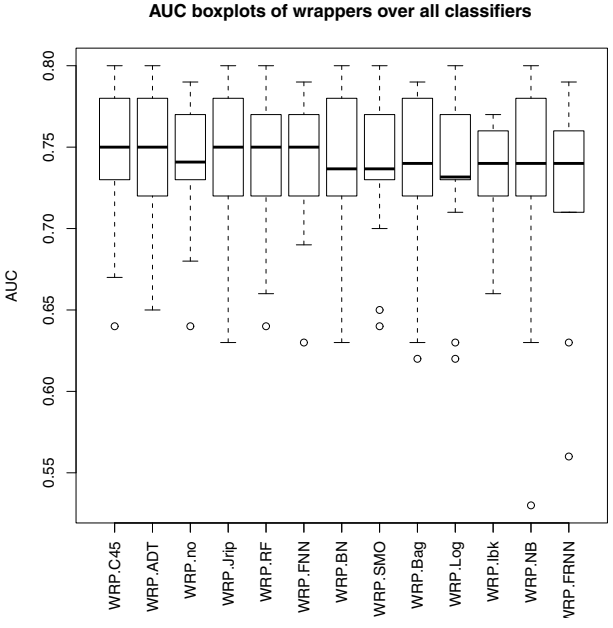


Fig. 1. Wrapper performance. AUC boxplots of the wrappers (including data without dimension reducing) over all classifiers ordered by decreasing AUC mean.

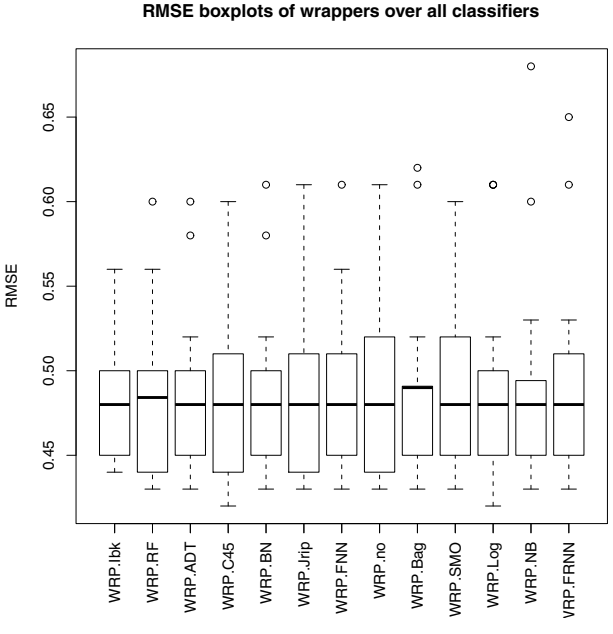


Fig. 2. Wrapper performance. RMSE boxplots of the wrappers (including data without dimension reducing) over all classifiers ordered by incresing RMSE mean.

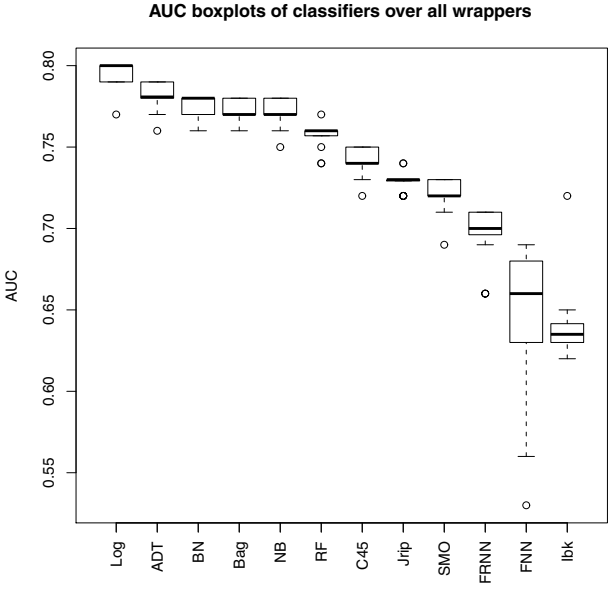


Fig. 3. Classifier performance. AUC boxplots of the classifiers over all wrappers (including data without dimension reducing) ordered by decreasing AUC mean.

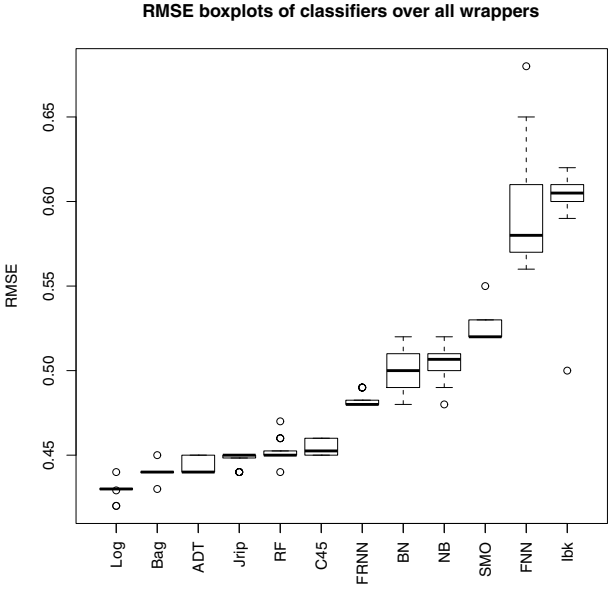


Fig. 4. Classifier performance. RMSE boxplots of the classifiers over all wrappers (including data without dimension reducing) ordered by decreasing RMSE mean.

In our study, we found that no significant difference exists between investigated feature selection methods. Most of the GA-based wrappers achieved results no worse than high-dimensional dataset (denoted by WRP.no).

The statistical results obtained and the multiple means comparison tests analyzed make the three classifiers ADT, Log and Bag competitive method to be considered in the field of animal breeding data mining, where there is a strong necessity of obtaining appropriate classification accuracy for dimension-reduced data. These classifiers obtained the highest mean rank when considering all the wrapper-based datasets and all the measures (AUC and RMSE). The statistical tests confirm that these differences are significant when these methods are compared to the other classifiers under study.

Particularly noteworthy is the fact that combination of a wrapper with the same classifier used as an induction method of the wrapper and next as a learning scheme over reduced set of attributes, gives average-good results in terms of AUC and RMSE.

4 Conclusions

In this work, we have conducted an extensive study of both classification methods as well as GA-based feature selection methods for classification of animal breeding data. The Weka software was used as a machine learning tool. Therefore, we can assume an almost equal quality of implementations and differences can be attributed to the methods themselves and not to implementations. The experiments focused on identifying the best combination of classifier and feature selection strategy.

In this study, we found that no significant difference exists between investigated feature selection methods. Most of the GA-based wrappers achieved results no worse than high-dimensional dataset (denoted by WRP.no).

The statistical results obtained and the multiple means comparison tests analyzed make the three classifiers ADT, Log and Bag competitive method to be considered in the field of animal breeding data mining, where there is a high necessity of obtaining appropriate classification accuracy for dimension-reduced data. These classifiers obtained the highest/lowest mean rank when considering all the wrapper-based datasets and all the measures (AUC and RMSE, respectively). The statistical tests confirm that these differences are significant when these methods are compared to the other classifiers under study. We are aware that the results drawn are biased by the selected dataset. You have to remember, that Silesian horse dataset is unique, both in terms of the horse species and attributes, and especially difficult would be to compare different wrappers over different - also in a term of dimensionality - datasets.

In a future work, we will investigate further classifiers also other feature selection methods (filters and embedded approaches) to perform more authoritative comparisons. It will be also necessary to explore more than one animal breeding dataset.

References

1. Bellman, R.: Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton (1961)
2. Burócziová, M., Řiha, J.: Horse breed discrimination using machine learning methods. J. Appl. Genet. 50(4), 375–377 (2009)
3. Caruana, R., Niculescu-Mizil, A.: Data mining in metric space: an empirical analysis of supervised learning performance criteria. In: Proceedings of the 10th Int. Conf. Knowl. Disc. Data Mining, pp. 69–78 (2004)
4. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874 (2006)
5. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 32, 674–701 (1937)
6. Garner, S.R., Holmes, G., McQueen, R.J., Witten, I.H.: Machine learning from agricultural databases: practice and experience. J. Computing 6(1a), 69–73 (1997)
7. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley (1989)
8. Jensen, R., Shen, Q.: Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches. IEEE Press, Wiley and Sons (2008)
9. Ling, C.X., Huang, J., Zhang, H.: AUC: A Better Measure than Accuracy in Comparing Learning Algorithms. In: Xiang, Y., Chaib-draa, B. (eds.) Canadian AI 2003. LNCS (LNAI), vol. 2671, pp. 329–341. Springer, Heidelberg (2003)
10. Walkowicz, E., Unold, O., Maciejewski, H., Skrobanek, P.: Zoometric indices in Silesian horses in the years 1945-2005. Ann. Anim. Sci. 11(4), 555–565 (2011)
11. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005)
12. Zhiwei, X., Xinghua, W.: Research for Information Extraction Based on Wrapper Model Algorithm. In: 2010 Second International Conference on Computer Research and Development, Kuala Lumpur, Malaysia, pp. 652–655 (2010)

A Simple Noise-Tolerant Abstraction Algorithm for Fast k-NN Classification

Stefanos Ougiaroglou* and Georgios Evangelidis

Dept. of Applied Informatics, University of Macedonia, 54006 Thessaloniki, Greece
{stoug,gevan}@uom.gr

Abstract. The k-Nearest Neighbor (k-NN) classifier is a widely-used and effective classification method. The main k-NN drawback is that it involves high computational cost when applied on large datasets. Many Data Reduction Techniques have been proposed in order to speed-up the classification process. However, their effectiveness depends on the level of noise in the data. This paper shows that the k-means clustering algorithm can be used as a noise-tolerant Data Reduction Technique. The conducted experimental study illustrates that if the reduced dataset includes the k-means centroids as representatives of the initial data, performance is not negatively affected as much by the addition of noise.

Keywords: k-NN classification, noisy data, clustering, data reduction.

1 Introduction

A supervised classification algorithm (or classifier) is a data mining method that assigns new data into one of a predefined set of categories (or classes) [13]. Classifiers can be either eager or lazy. Eager classifiers initially build a classification model, which is then used to classify new items. In contrast, lazy classifiers make classification predictions by scanning the available data when a new item arrives.

The k-Nearest Neighbor (k-NN) classifier is a typical example of a lazy classifier [6]. It classifies new items by a majority vote of their nearest neighbors. A new item is assigned into the class that is most common amongst its k nearest neighbors. The closeness of two items is defined using a distance metric (e.g., Euclidean distance). Ties are resolved either by choosing the class of the nearest neighbor or randomly. In general, k-NN is a simple and easy to implement classifier, is considered to be effective and has many applications.

Despite all its advantages, the k-NN classifier involves the major drawback of high computational cost: all distances between a new unclassified item and the training items need to be computed. This weakness constitutes an active research field and has attracted the interest of researchers from different areas (databases, data mining, machine learning). Therefore, many speed-up searching methods have been proposed. They are mainly based on Indexing [29,22], Data Reduction Techniques (DRTs) [25,11] and Cluster-based methods [30,16,26]. Indexing and

* S. Ougiaroglou is supported by the State Scholarship Foundation of Greece (I.K.Y.).

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 210–221, 2012.

Cluster-based methods, for each new item, prune the multidimensional space and search for nearest neighbors within a dynamically formed subset of the initial data. DRTs have the extra benefit of storage reduction. This work focuses on DRTs, and particularly on how their performance is affected by the addition of noise in the training data.

DRTs can be divided into two major algorithm categories: (i) abstraction (or generation) algorithms [25] that generate new items to represent the initial training set, and, (ii) filtering (or selection) algorithms [11] that keep some items from the initial training set as representatives. Both aim to build a small representative set of the initial training data. Applying the k-NN classifier using the reduced (or condensing) set one can achieve almost the same accuracy as when using the original dataset, and also, one has the advantage of much lower computational cost and storage requirements.

Many filtering algorithms try to to keep only the close-class-border items of the initial training dataset. The idea is that the "internal" items of a class can be removed without loss of accuracy, since they do not define boundaries among classes. Similarly, abstraction algorithms generate a few representatives for the "clear" areas (such as the "internal" class area) and more representatives for the close-border areas. A subcategory of filtering algorithms are the editing approaches. They aim to increase the accuracy rather than speed-up the classification process. This is achieved by filtering the noisy items and smoothing the decision boundaries (removing close-class-border items) of the training data. Finally, some DRTs are characterized as hybrid (see [11]), since they incorporate an editing mechanism into the main reduction procedure.

The size of the condensing set, and, therefore, the effectiveness of data reduction, depends on the level of noise in the data as well as the number of discrete classes (the more the classes the more the close-class-border items selected or generated). Consequently, a complete preprocessing step may include an editing procedure to remove the noise from the data. However, editing may be inappropriate because either it may be not able to remove all noisy items or it may remove useful data. Moreover, editing constitutes an extra, costly preprocessing step that may be unacceptable in some application domains. For instance, the well-known Edited Nearest Neighbor (ENN) rule [28] needs to compute $\frac{N*(N-1)}{2}$ distances in order to edit a training set of N items. Thus, there is a need for a noise-resistant or noise-tolerant algorithm. These observations are behind the motivation of this work.

The main goals of this paper are: (i) to examine how the addition of noise affects the performance of two state-of-the-art DRTs, the filtering method CNN-rule [14] and the abstraction approach RSP3 [23], and, (ii) to propose the use of the centroids produced by k-means clustering [19] on the training sub-datasets belonging to each class as a simple, noise-tolerant approach.

The rest of this paper is organised as follows. Section 2 considers in detail Hart's condensing as well as the family of RSP algorithms. In Section 3, the condensing through k-means algorithm is presented. Finally, Section 4 presents the experimental results, and Section 5 concludes the paper.

2 State-of-the-Art Abstraction and Filtering Algorithms

Data reduction for speeding-up the k-NN classification has attracted many researchers since the late 60s. Many of the proposed algorithms have been implemented under the KEEL software [2] which is an open-source Java-based framework. Reviews, taxonomies, evaluations and comparisons of the abstraction and filtering algorithms can be found in [25] and [11], respectively. Other reviews on DRTs can be found in [18,24,27]. We focus on two well-known methods: (i) CNN-rule [14] and (ii) the family of RSP algorithms [23].

2.1 Condensing Nearest Neighbor Rule

The first and a well-known data reduction algorithm has been proposed by Hart. It is the Condensing Nearest Neighbor (CNN) rule [14] and it belongs to the filtering category. Although many other approaches extend or are based on it, CNN-rule remains the condensing algorithm of reference until today. Some variations include the Reduced NN (RNN) Rule [12], the Selective NN (SNN) Rule [21], the Modified CNN (MCNN) Rule [8], and, the Fast CNN (FCNN) Rule [4].

Like many other DRTs, CNN-rule tries to remove the non-close-class-border items from the initial training set. This is achieved by using two lists, S and T. First, an item of the training set is put in S and all other items are put in T. Then, the CNN-rule attempts to classify the items of T by scanning the items of S and using the 1-NN approach. When an item is misclassified, it is moved to S. The procedure stops when there is no move from T to S during a pass of T. The items that have been placed in S constitute the condensing set.

The aforementioned procedure is based on the idea that if an item is wrongly classified, it is probably near to the decision boundaries and so it must be placed into the condensing set. Contrary to other approaches, CNN-rule determines the size of the condensing set automatically.

2.2 Reduction by Space Partitioning

A popular abstraction algorithm has been proposed by Chen and Jzwik [5]. The algorithm divides the training set into groups by considering the distance between the most distant items in the groups (i.e., diameter). More specifically, it initially retrieves the pair of the most distant items, A and B in the training set T. Then, T is divided into two groups T_A and T_B. The items of T that are closer to A (B) are placed in T_A (T_B). This procedure continues recursively for each group and terminates when a user-defined number of groups are created. Finally, for each group, the algorithm finds the most common class and computes a mean item assigned to this class. These items constitute the condensing set.

The Reduction by Space Partitioning family of algorithms (RSP) [23] is based on the aforementioned method. Particularly, the RSP family introduces three algorithms. RSP1 computes for each group as many mean items as the different classes in the group. For instance, if one group includes items from two different classes, the algorithm will compute two mean items for this group. Although

this approach leads to a larger condensing set than that of the Chen and Jzwik algorithm, it takes into account all items of the initial training-set and aspires to achieve better classification accuracy.

RSP1 and RSP2 differ on the way they select the next group to be divided. RSP1 selects the group with the largest diameter based on the idea that this group may include more items than the other groups, and, thus, a highest reduction can be achieved. RSP2 uses as a splitting criterion the overlapping degree of each group. The group with the highest overlapping degree is split first. RSP2 is based on the theory that the items that belong to a specific class should be as close to each other as possible.

RSP3 does not use a splitting criterion. All non homogeneous groups are split until they do not contain items from different classes. RSP3 is the only RSP method that determines automatically (without user-defined parameters) the number of items that constitute the condensing set. Furthermore, RSP3, like many other abstraction approaches, generates few items for representing the non close-class-border areas, and many more items for the close-class-border areas.

3 Reduction through *k*-Means Clustering

A simple but effective abstraction approach could use k-means clustering. More specifically, we propose the use of the k-means centroids for the construction of the condensing set. Of course, there are many approaches that use clustering either to summarize similar items [7,15] or to condense the initial training set [20,9,3] for speed-up purposes. Furthermore, there are cluster-based methods that although do not reduce the storage requirements, they are able to speed-up the k-NN classifier [16,30,17,26] by excluding clusters form the nearest neighbors searching procedure. In this paper, we focus on the noise tolerance of the abstraction/filtering algorithms and our purpose is to ascertain if the k-means approach can improve the classification performance. We term this approach Reduction through k-means clustering (RkM).

RkM involves a clustering preprocessing step on the training dataset. Particularly, k-means is executed on the items of each class. Thus, for each class, a number of clusters is identified and the class is represented by the mean-vectors (or centroids) of these clusters. Considering this simple abstraction approach, it is clear that noisy items of a class are represented by a cluster centroid lying in the main class area of this class. Thus, we suppose that RkM is a more noise-tolerance reduction method and its effectiveness does not depend as much on an editing procedure as CNN-rule and RSP3 (and many other DRTs).

An issue that needs to be addressed is the determination of the number of centroids that should represent each class. We deal with this issue by introducing a parameter called Data Reduction Factor (DRF). For each class C, RkM builds $\lceil \frac{|C|}{DRF} \rceil$ clusters, where $|C|$ is the number of items that belong to C. The use of ceiling ensures that each class will be represented by at least one centroid. Thus, in the RkM condensing set are placed only the centroids produced by the k-means clustering. The DRF parameter allows the user to determine the desirable trade-off between accuracy and computational cost. Particularly, low DRF values lead

Algorithm 1. Reduction through k-means clustering

Input: *DRF, Training-set* **Output:** *Condensing-set*

```
1: for each class C of Training-set do
2:    |C| ← count of the items of Training-set that belong to C
3:    NC ← ⌈|C| / DRF⌉
4:    Use the first NC items of C as the initial means
5:    Initialize the current cluster of each item t_i of C to be NULL
6:    repeat
7:       item_has_moved ← false
8:       for each item t_i of C do
9:          Find the nearest cluster candidate_cluster to t_i (according to its mean)
10:         if candidate_cluster ≠ current cluster of t_i then
11:            Assign t_i to candidate_cluster
12:            item_has_moved ← true
13:         end if
14:      end for
15:      Compute the new mean for each cluster of C
16:   until item_has_moved = false {no item has moved}
17:   Place the mean vectors of C into the Condensing-set
18: end for
19: return Condensing-set
```

to accurate classifiers with small gains in execution cost, whereas, high DRF values build fast classifiers that achieve lower accuracy. The RkM procedure is outlined in Algorithm 1. Effectively, RkM applies k-means clustering on all classes of the training dataset.

4 Performance Evaluation

4.1 Experimental Setup

The filtering/abstraction algorithms presented in Sections 2 and 3 were implemented in C and tested against each other using six real life datasets summarized in Table 1. The first four datasets were retrieved from the UCI Machine Learning Repository [10], while the Texture and Phoneme datasets from the KEEL Repository [1]. We split LR, MGT, TXR and PH into training and testing sets using a random sampling procedure. Concerning LS and PEN, we used the training/testing splits defined in the UCI Repository.

To evaluate the performance of the algorithms on noisy conditions, eight versions of LR, LS, PEN, and TXR datasets with varying degree of noise (from 0% to 70% with step=10) were constructed and tested. The first version was the original dataset (without extra noise) whereas the last version contained 70% noisy data. MGT and PH datasets have only two classes and so they can not afford high levels of additional noise. Thus, we used four versions for these datasets with noise ranging from 0% to 30%. Note that, MGT already includes a considerably high level of noise in its original form.

Table 1. Datasets description

Dataset	Training items	Testing items	Attr.	Classes
Letter Recognition (LR)	15000	5000	16	26
Landsat Satellite (LS)	4435	2000	36	6
Pendigits (PEN)	7494	3498	16	10
Magic Gamma Telescope (MGT)	14000	5020	10	2
Texture (TXR)	4400	1100	40	11
Phoneme (PH)	4000	1404	5	2

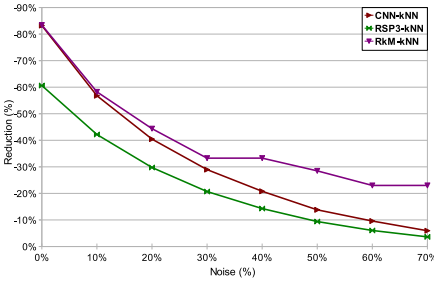
The noisy versions of all datasets were constructed by adding random noise of a specific probability to the datasets. For instance, when a dataset with 20% noisy data was needed, the class attribute of each item of the corresponding training set was modified by selecting another random class-attribute with a probability of 0.2. Thus, for each tested algorithm, eight or four pairs of performance measurements (reduction rate and classification accuracy) were obtained. For each dataset, we present a pair of figures (Fig. 1–6): figures (a) demonstrate the reduction rates of each method, whereas, figures (b) show the corresponding accuracy values. Figures (b) include an extra curve for the conventional k-NN (classification by scanning the original training data). Finally, we note that the datasets were used without data normalization or any other data transformation, and that the adopted distance metric was the Euclidean distance.

We compared the performance of the RkM abstraction algorithm to that of CNN-rule and RSP3 that are good representatives of the two different algorithm categories, i.e., filtering and abstraction algorithms. Initially, each one of these algorithms was executed in order to produce the corresponding condensing set by scanning the initial training data. Then, for each item of the testing set, the k-NN classifier made a classification prediction by searching for nearest neighbors over the condensing set produced. We refer to these k-NN classifiers as CNN-kNN, RSP3-kNN and RkM-kNN. The chosen k parameter values for all the aforementioned classifiers were the ones that achieved the highest accuracy.

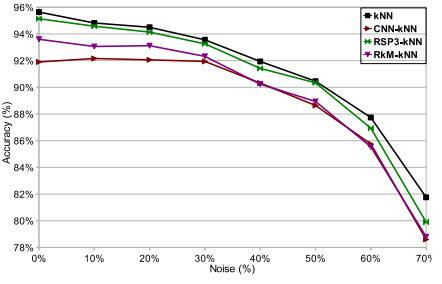
For RkM, recall that DRF determines the number of representatives that will be created for each one class. The value of this parameter can be used to adjust the cost vs. accuracy trade-off required by the application domain. In our experiments, we wanted to build RkM classifiers that would achieve accuracy comparable to that of CNN-k-NN and RSP3-kNN. In particular, in most cases, DRF values were appropriately adjusted to achieve an RkM-kNN accuracy between the corresponding accuracy values of CNN-kNN and RSP3-kNN.

4.2 Comparisons

As we expected, the experimental measurements illustrated in Figures 1–6 demonstrate that RkM can be characterized as a more noise-tolerant approach. The preprocessing procedures of CNN-rule and RSP3 are not able to reduce as much the size of the condensing set when the initial training set contains noise. In the

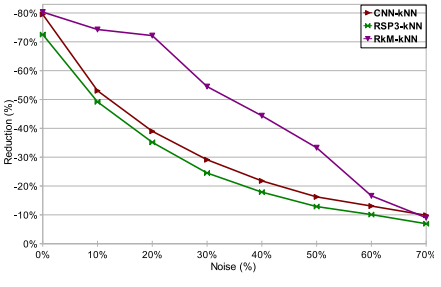


(a) Reduction rate

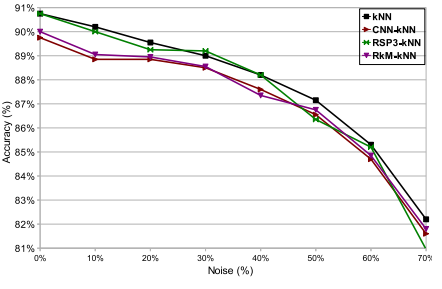


(b) Accuracy

Fig. 1. Letter Recognition dataset



(a) Reduction rate



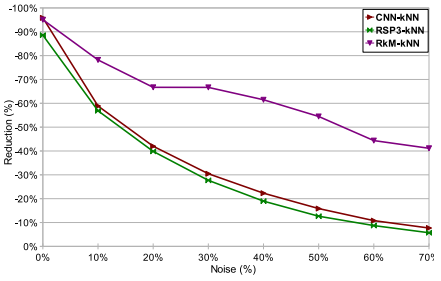
(b) Accuracy

Fig. 2. Landsat Satellite dataset

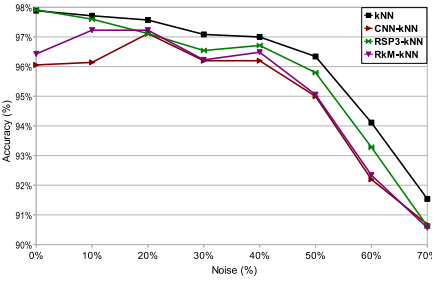
case of CNN-rule, this is because more items are misclassified and so, they are moved to the condensing set. In the case of RSP3, the algorithm keeps on splitting the many non-homogeneous groups that are produced. On the other hand, RkM does not seem to be affected as much by the addition of noise. Almost in all cases, CNN-rule achieved higher reduction rates than RSP3 and RSP3-kNN achieved higher accuracy values than CNN-kNN.

RkM reduction rates were higher in the cases of LS (Fig. 2), PEN (Fig. 3) and PH (Fig. 6) than that of LR (Fig. 1) and TXR (Fig. 5). This is because, in the later datasets, CNN-kNN and RSP3-kNN achieved high accuracy values that RkM-kNN can not easily achieve. Consequently, RkM classifiers that consist of more representatives (lower DRF values) are required in order to achieve comparable performance. However, even in the cases of LR and TXR, it is obvious that the reduction rates of RkM are higher than that of the other two methods.

In the case of MGT, RkM-k-NN did not perform as good. However, in the original version of MGT, which already includes many noisy items, RkM had higher reduction rates than CNN-rule and RSP3 achieving the same accuracy as CNN-kNN. Finally, in many cases, RkM-k-NN achieved higher accuracy values than those we present here. However, they involved lower reduction rates. Our purpose was to ascertain whether RkM is a noise-tolerant approach and so, we

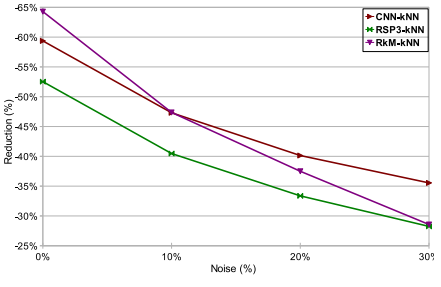


(a) Reduction rate

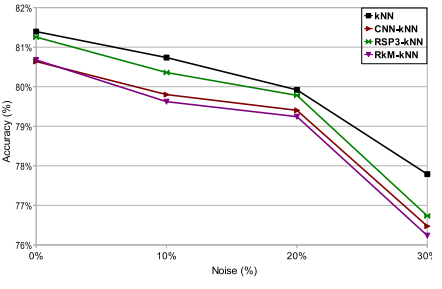


(b) Accuracy

Fig. 3. Pendigits dataset



(a) Reduction rate



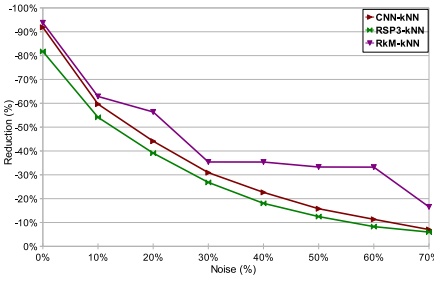
(b) Accuracy

Fig. 4. Magic Gamma Telescope dataset

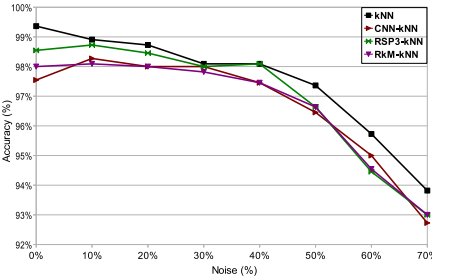
focused on the reduction rates that it achieved. The user of RkM can define the desirable cost vs. accuracy trade-off through the DRF parameter.

RkM performed very well on edited datasets. We repeated the experiments on LR, PEN, and LS after first editing the versions of each dataset in order to remove the noisy data. For editing purposes, we used the ENN-rule with k=3 [28]. RkM managed to reach or exceed the reduction rates of CNN-rule and RSP3, while achieving comparable accuracy. Figure 7 shows the results obtained on the LS dataset. Finally, we should mention that the accuracy measurements presented may have been slightly different had we used different training/testing splits. However, we are mainly interested in the reduction rates that the presented methods achieved and so, the slightly different accuracy measurements are not critical.

Concluding this section, we focus on the preprocessing cost needed for the construction of the condensing sets of the presented methods. Of course, these cost measurements are relevant only once at the beginning of the data mining process. However, there are many application domains that periodically accept new training items and so the reconstruction of the condensing set may be inevitable (incremental approaches are not always applicable). Thus, the preprocessing cost may be a critical issue.

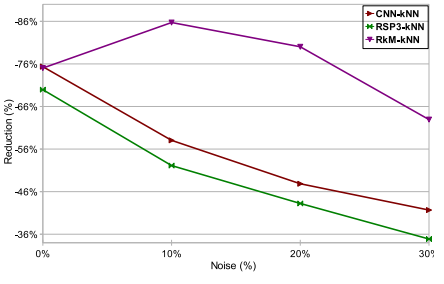


(a) Reduction rate

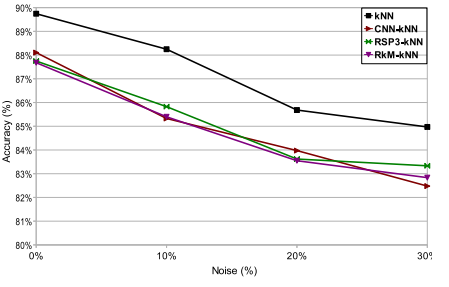


(b) Accuracy

Fig. 5. Texture dataset

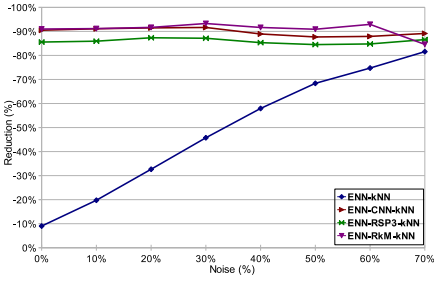


(a) Reduction rate

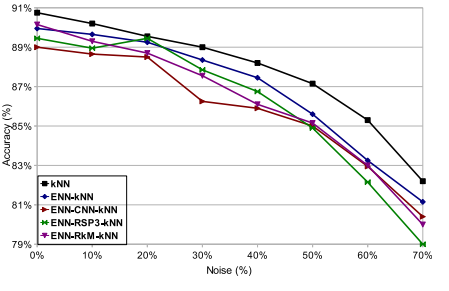


(b) Accuracy

Fig. 6. Phoneme dataset



(a) Reduction rate



(b) Accuracy

Fig. 7. Edited Landsat Satellite dataset

Table 2 presents these measurements for each original dataset in terms of distance computations. Considering the results, we conclude that RSP3 is the most expensive approach. This is because RSP3 includes a function that finds the most distant items in each group. The preprocessing cost of RkM depends on the DRF value. The last column of Table 2 lists the DRF values that we used in order to build the RkM classifiers presented in Figures 1–6 (noise=0%).

Table 2. Preprocessing cost (distance computations)

Dataset	CNN-rule	RSP3	RkM	DRF
Letter Recognition	145,386,010	291,151,380	11,576,185	6
Landsat Satellite	13,545,272	28,929,950	5,771,993	5.1
Pendigits	7,940,953	70,561,629	4,409,382	20
Magic Gamma Telescope	217,900,759	412,752,916	323,718,012	2.8
Texture	5,189,518	25,361,045	1,278,585	18
Phoneme	13,532,827	17,847,352	22,809,139	4

For LR, LS, PEN and TXR, RkM was a cheaper approach than the other methods. On the other hand, for MGT and PH, its cost was considerably higher. If the application domain does not afford the RkM preprocessing step, it can be sped-up by adopting a more efficient stop condition for k-means than the complete cluster consolidation (see Algorithm 1).

5 Conclusion

We examined how the state-of-the-art algorithms CNN-rule and RSP3 are negatively affected by the addition of noise in the data. In addition, we demonstrated that the the well-known k-means clustering can be used as a noise-tolerant data reduction method. The experimental results showed that the reduction through k-means (RkM) algorithm can be used to achieve comparable accuracy to CNN-rule and RSP3 but with much higher data reduction rates, especially on very noisy datasets. Even on datasets without noise, RkM can compete with CNN-rule and RSP3 in terms of data reduction rates and classification accuracy.

We plan to keep on examining the way clustering algorithms can be used for effective and fast k nearest neighbor classification. Particularly, our future work includes the use of clustering approaches in order to either reduce the training data or devise algorithms that search for nearest neighbors in an on-line subset of the data formed at the time that a new, unclassified item arrives.

References

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Multiple-Valued Logic and Soft Computing 17(2-3), 255–287 (2011)
2. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesús, 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 Comput. 13(3), 307–318 (2009)
3. Alizadeh, H., Minaei-Bidgoli, B., Amirgholipour, S.K.: A new method for improving the performance of k nearest neighbor using clustering technique. JCIT 4(2), 84–92 (2009)

4. Angiulli, F.: Fast condensed nearest neighbor rule. In: Proceedings of the 22nd International Conference on Machine Learning, ICML 2005, pp. 25–32. ACM, New York (2005)
5. Chen, C.H., Jóźwik, A.: A sample set condensation algorithm for the class sensitive artificial neural network. Pattern Recogn. Lett. 17, 819–823 (1996)
6. Dasarathy, B.V.: Nearest neighbor (NN) norms : NN pattern classification techniques. IEEE Computer Society Press (1991)
7. Datta, P., Kibler, D.: Learning symbolic prototypes. In: Proceedings of the Fourteenth ICML, pp. 158–166. Morgan Kaufmann (1997)
8. Devi, V.S., Murty, M.N.: An incremental prototype set building technique. Pattern Recognition 35 (2002)
9. Eick, C.F., Zeidat, N.M., Vilalta, R.: Using representative-based clustering for nearest neighbor dataset editing. In: ICDM, pp. 375–378 (2004)
10. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
11. Garcia, S., Derrac, J., Cano, J., Herrera, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Transactions on Pattern Analysis and Machine Intelligence 99(prePrints) (2011)
12. Gates, G.W.: The reduced nearest neighbor rule. IEEE Transactions on Information Theory 18(3), 431–433 (1972)
13. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science (2011)
14. Hart, P.E.: The condensed nearest neighbor rule. IEEE Transactions on Information Theory 14(3), 515–516 (1968)
15. Hruschka, E.R., Hruschka, E.R.J., Ebecken, N.F.: Towards efficient imputation by nearest-neighbors: A clustering-based approach. In: Australian Conference on Artificial Intelligence, pp. 513–525 (2004)
16. Hwang, S., Cho, S.: Clustering-Based Reference Set Reduction for k-nearest Neighbor. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007, Part I. LNCS, vol. 4492, pp. 880–888. Springer, Heidelberg (2007)
17. Karamitopoulos, L., Evangelidis, G.: Cluster-based similarity search in time series. In: Proceedings of the 2009 Fourth Balkan Conference in Informatics, BCI 2009, pp. 113–118. IEEE Computer Society, Washington, DC, USA (2009)
18. Lozano, M.: Data Reduction Techniques in Classification processes (Phd Thesis). Universitat Jaume I (2007)
19. McQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proc. of 5th Berkeley Symp. on Math. Statistics and Probability, pp. 281–298. University of California Press, Berkeley (1967)
20. Olvera-Lopez, J.A., Carrasco-Ochoa, J.A., Trinidad, J.F.M.: A new fast prototype selection method based on clustering. Pattern Anal. Appl. 13(2), 131–141 (2010)
21. Ritter, G., Woodruff, H., Lowry, S., Isenhour, T.: An algorithm for a selective nearest neighbor decision rule. IEEE Trans. on Inf. Theory 21(6), 665–669 (1975)
22. Samet, H.: Foundations of multidimensional and metric data structures. The Morgan Kaufmann series in computer graphics. Elsevier,Morgan Kaufmann (2006)
23. Sánchez, J.S.: High training set size reduction by space partitioning and prototype abstraction. Pattern Recognition 37(7), 1561–1564 (2004)
24. Toussaint, G.: Proximity graphs for nearest neighbor decision rules: Recent progress. In: 34th Symposium on the INTERFACE, pp. 17–20 (2002)
25. Triguero, I., Derrac, J., García, S., Herrera, F.: A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE Transactions on Systems, Man, and Cybernetics, Part C 42(1), 86–100 (2012)

26. Wang, X.: A fast exact k-nearest neighbors algorithm for high dimensional search using k-means clustering and triangle inequality. In: The 2011 International Joint Conference on Neural Networks (IJCNN), pp. 1293–1299 (August 2011)
27. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning 38(3), 257–286 (2000)
28. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. on Systems, Man, and Cybernetics 2(3), 408–421 (1972)
29. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search - The Metric Space Approach, vol. 32. Springer, Heidelberg (2006)
30. Zhang, B., Srihari, S.N.: Fast k-nearest neighbor classification using cluster-based trees. IEEE Trans. Pattern Anal. Mach. Intell. 26(4), 525–528 (2004)

Adaptive Inventory Control in Production Systems

Balázs Lénárt[1], Katarzyna Grzybowska[2], and Mónika Cimer[1]

[1] Budapest University of Technology and Economics, Department of Transport Technology, Műegyetem Rakpart 3, 1111 Budapest, Hungary
[2] Poznan University of Technology, Faculty of Engineering Management, Chair of Production Engineering and Logistics, Strzelecka 11, 60-965 Poznan, Poland

Abstract. This paper presents an adaptive inventory control system based on neuro-fuzzy logic. In particular we describe a control system using adaptive neuro-fuzzy interference (ANFIS) for calculating the optimal value of the storage level of goods. An implementation in MATLAB had been used to test and verify the proposed idea for the economic order quantity as a simple inventory control system. Our results shows that the presented approach is able to determine the optimal stock level and cost without knowing the exact mathematical model of the examined system.

Keywords: Logistics, Inventory control, Neuro-Fuzzy, ANFIS.

1 Introduction

The production and the actual usage of a product mostly differ in time and space resulting into a disorder of the inventory system during the distribution of the goods. To achieve and preserve a smooth supply chain process it is crucial to maintain a given level of storage. Furthermore, the optimal value of this stock level has to be maintained due to incidental expenses such as worth and stock holding costs. This could be difficult as corporate processes are in most cases stochastic and inordinately complex. There have been proposals for adaptive inventory control systems [1] using mainly soft computing [2] [3] and simulation techniques [4] [5].

The aim of the current research is to demonstrate that a neuro-fuzzy [6] based logic controller is capable of solving complex inventory problems and with future improvements it is able to achieve coverage of all important parameters of the target system. We implemented such a system based on the ANFIS [7] method. It is basically a decision support system with neuro-fuzzy rules therefore it is able to provide the order scheduling parameters of goods at operational level. The paper is structured as follows: Section 2 gives a general overview about inventory control systems while Section 3 explains the application of the ANFIS controller interface. In Section 4 we describe the simulation setup whereas the results are presented in Section 5. This paper is concluded in Section 6.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 222–228, 2012.

2 Inventory Control

The function of an inventory control system (Figure 1) is to fulfil the demand of the current production or consumption process, to operate inventory strategies and to keep the optimal stock level [8]. Accordingly such control systems can be divided into a stock monitoring, an inventory strategy (policy) and an order placing subsystem. The controller is directly connected to the examined process and through order placing it influences the stock levels in various parts of the supply chain.

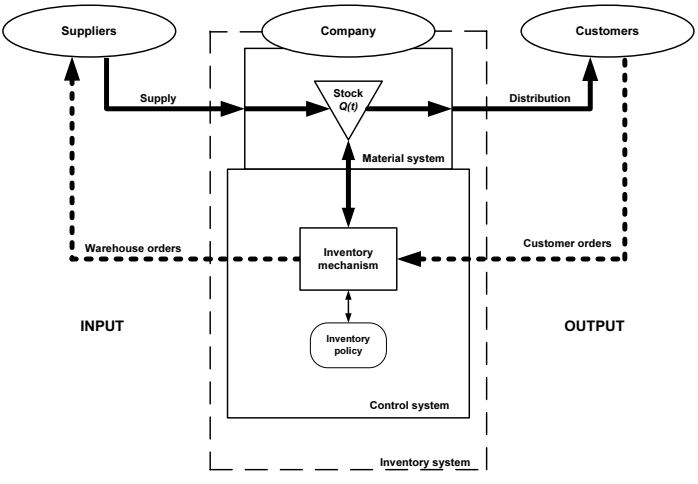


Fig. 1. Inventory control system

Such systems can be controlled by various decision support systems to achieve an optimal workflow of goods by solving problems occurring during inventory processes. However this highly adaptive system acts properly for time series with constant expected values, but in real conditions the demand data contains dynamic elements such as trend, season and white noise [9]. Considering these effects an extended concept with feedback was developed (Figure 2).

3 The ANFIS Controller

We propose the application of an Artificial Neural-Fuzzy Interface (ANFIS) as a basis of the inventory control system. The learning capabilities are provided by a SISO (Single Input - Single Output) artificial neural network. The set of input values are determined by the expectation values of the demand. The required fuzzy sets are 11 Gauss-functions to cover the base set of the mean values; accordingly the rule set of the fuzzy system contains 11 "IF-THEN" rules [10].

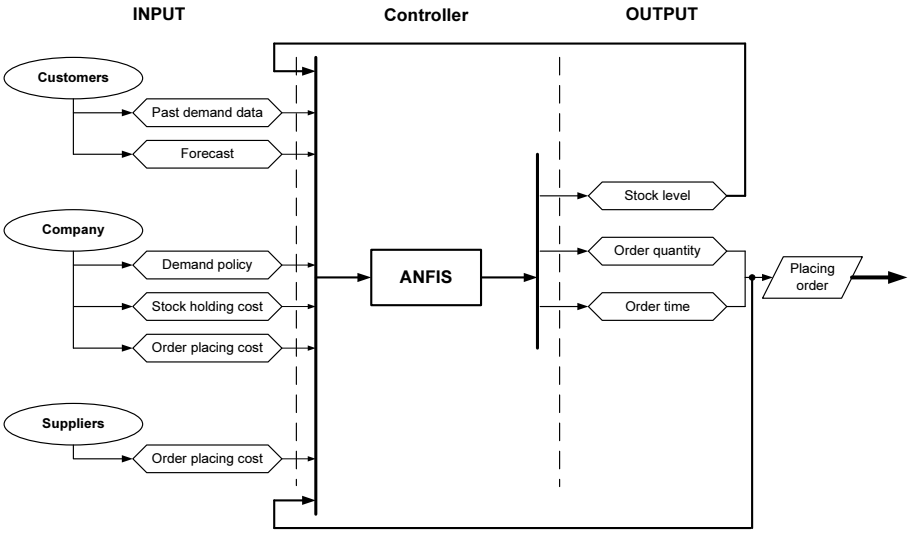


Fig. 2. Concept of an adaptive inventory control system based on ANFIS with feedback

Training of the network is done by feeding earlier period demand data. With the aggregation of the results of the fuzzy rules the output of the network can be calculated, which is currently the optimal order quantity. This order quantity is a minimum value of a cost function (Figure 3) and describes a parameter set of the optimal values.

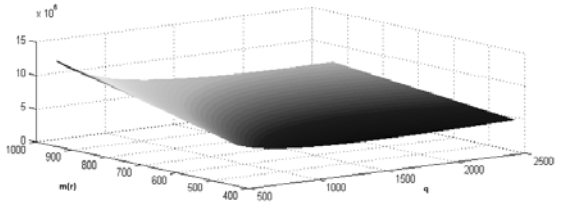


Fig. 3. Cost function

4 Simulation Setup

For testing and validation purposes we implemented the proposed approach using MATLAB and the GUI builder component. On the graphical user interface input data such as costs, examined period, expectation values of the demand in

addition the earlier period demand data can be maintained. Excel spreadsheet is being used as input source, but an interface is being planned to connect the controller to an existing SQL database or enterprise resource planning (ERP) system. The program then calculates the results - such as optimal order quantity and interval and plots them on the screen. It is relevant to note that the training of the time series has to be executed only once in the initialization phase. In our simulation we used the economic order quantity (EOQ) [11] - a simple inventory model to verify our system. The input dataset, namely the earlier period demand data was characterized by normal distribution, different expectation values and variances. In some cases the costs were also dynamically changing.

5 Results

After evaluating different datasets we obtained the following results. Table 1 depicts the clustered expectation values of the earlier period demand data and variances used as input data. The third column of the table contains the difference between the predicted an expected values. The results clearly show that in

Table 1. Results

Expectation values	Variance	Difference [%]
200...500	10 %	0
500...800	10 %	0
500...800	20 %	0
800...1100	5 %	0
800...1100	10 %	0
800...1100	20 %	0
800...1100	30 %	0

all of the examined cases the controller provided the optimal value. Conclusively the ANFIS driven inventory control system proved to be capable of accomplishing the above defined simple inventory control tasks successfully. Thus, further tests are required to check the system against non-static, more complex input datasets where the time series contain trend and seasonality tags as well. In the simulation multiple demand trigger data ($r(t)$) was used as input. The datasets contain a 60 days long data series with linear and non linear scenarios. During the simulation the time interval is divided into 10 days long periods, so the appropriate input of the ANFIS is the sum of customer demand data for the actual 10 days:

$$R_{10} = \sum_{i=1}^{10} r(t_i) \quad [items] \tag{1}$$

The optimal order quantity and time is determined by the aggregated (R_{10}) function and at the calculated time the order can be placed, but at the next

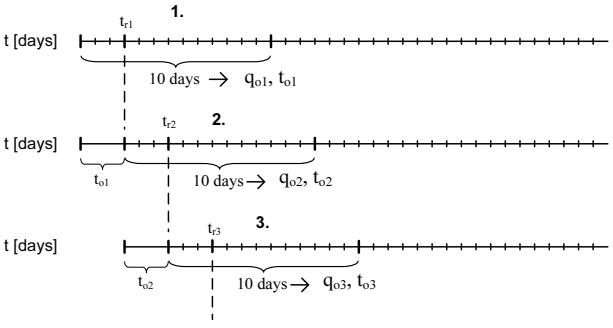


Fig. 4. Order placing times

order time a re-optimization is being made for the upcoming 10 days period. This method is illustrated on (Figure 4). By examining the basic EOQ concept it would seem that during the simulation of 60 days at some re-order days the stock level did not explain the new order, however it was placed and caused overstocking. In some cases the opposite occurred: - because of the predefined order time - reorder could not be performed so stock shortage has taken place (Figure 5, EOQ column). To avoid this phenomena at the re-order time the stock curve must tend to the $Q(t) = 0$ straight line since at order time the $Q(tr)$ value must be in the $[0,\varepsilon]$ interval, where $Q(t)$ is the actual stock level and ε is an arbitrarily small positive number. This can be achieved by connecting the output parameters such as stock level, order quantity and time of the controller back to the input (Figure 2). With this feedback the controller can constantly follow the changes of the stock level and can initialize a new parameter set if the current optimal order time is not satisfactory.

The system will intervene into the inventory process:

- if the inventory level fell below zero in the following term would cause stock shortage;
- if the aggregate demand in the following term is below the current inventory level preventing overstocking.

Accordingly the controlled process provides a curve without the unwanted effects shown on Figure 5 A-EOQ (Adaptive-EOQ) column. As a result it is clearly visible that the net signed area bounded by the graph of inventory level is decreased compared to the area of the original concept in the EOQ column. The working capital has also decreased. In this manner an inventory control system was developed which can determine the optimal order parameter set independently from the type of the demand time series, also generates savings in the inventory cost.

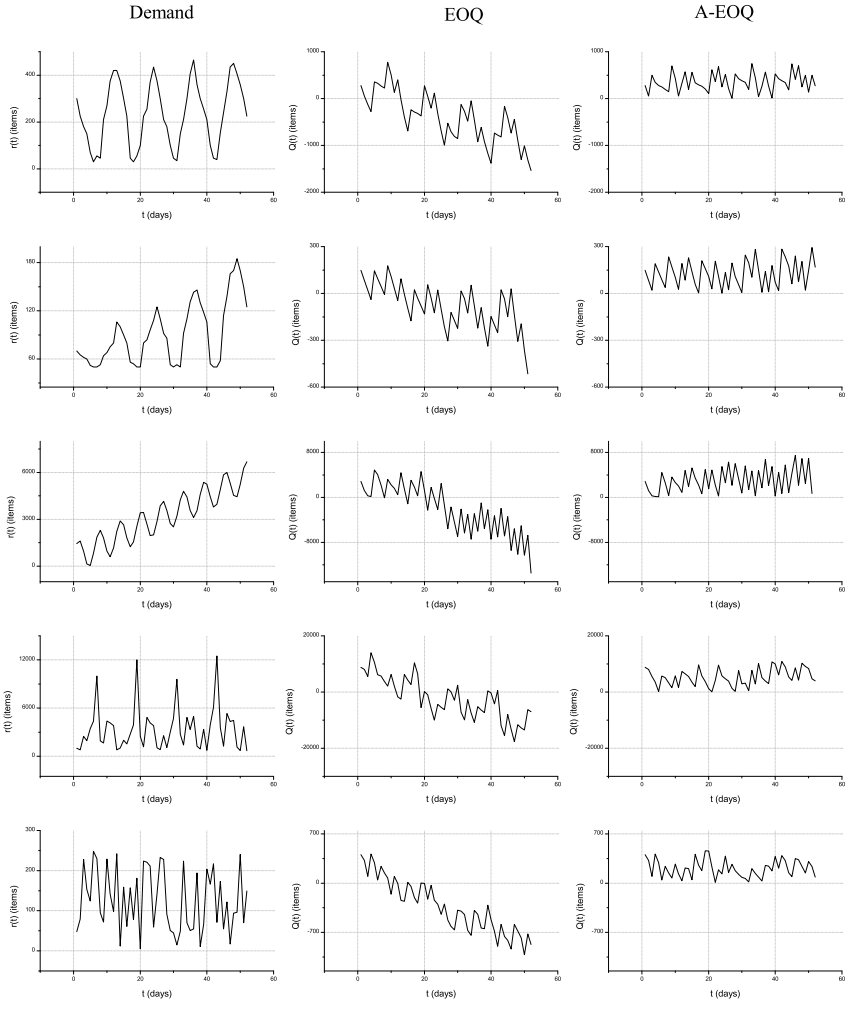


Fig. 5. Results

6 Conclusion

One possible way to improve current inventory control system is the usage of artificial intelligence methods such as neural network, fuzzy system and genetic algorithm since a human-made decision support system (DSS) can behave similar way as an intelligent living being. It means that is less sensitive to input errors through its intuitive capability. The concept of applying adaptive neuro-fuzzy inventory controller based on ANFIS had been proposed and implemented in simulation. A basic version was designed and the usage of soft computing techniques like ANFIS was validated in solving complex inventory control tasks. Future research is planned in this area with extended parameters and capabilities furthermore applying novel neuro-fuzzy methods.

Acknowledgement. This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Széchenyi Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

References

1. Axsäter, S.: Inventory control. Springer, Heidelberg (2006)
2. Rezaei, J., Davoodi, M.: Genetic Algorithm for Inventory Lot-Sizing with Supplier Selection Under Fuzzy Demand and Costs. In: Ali, M., Dapoigny, R. (eds.) IEA/AIE 2006. LNCS (LNAI), vol. 4031, pp. 1100–1110. Springer, Heidelberg (2006)
3. Rezaei, J., Davoodi, M.: Genetic Algorithm for Inventory Lot-Sizing with Supplier Selection Under Fuzzy Demand and Costs. In: Ali, M., Dapoigny, R. (eds.) IEA/AIE 2006. LNCS (LNAI), vol. 4031, pp. 1100–1110. Springer, Heidelberg (2006)
4. Bóna, K.: Simulation supported optimisation of inventory control processes by application of genetic algorithms. In: European Simulation Symposium, Budapest (2004)
5. Schwartza, J., Wanga, W., Rivera, D.: Simulation-based optimization of process control policies for inventorymanagement in supply chains. Automatica 42(8), 1311–1320 (2006)
6. Jin, Y.: Fuzzy modelling of high-dimensional systems: Complexity reduction and interpretability improvement. IEEE Transactions on Fuzzy Systems 8(2), 212–221 (2000)
7. Nedjah, N.: Studies in Fuzziness and Soft Computing, pp. 53–83. Springer, Heidelberg
8. Astanti, D.R., Luon, T.H.: A heuristic technique for inventory replenishment policy with increasing demand pattern and shortage allowance. The International Journal of Advanced Manufacturing Technology 41(11-12), 1199–1207 (2008)
9. Wagner, N., Michalewicz, Z., et al.: Intelligent techniques for forecasting multiple time series in real-world systems. International Journal of Intelligent Computing and Cybernetics 4(3), 284–310 (2011)
10. Rotshtein, A.P., Rakityanskaya, A.B.: Inventory control as an identification problem based on fuzzy logic. Cybernetics and Systems Analysis 42(3), 411–419 (2006)
11. Wilson, R.H.: A Scientific Routine for Stock Control. Harvard Business Review 13, 116–128 (1934)

Hybrid Artificial Intelligence System in Constraint Based Scheduling of Integrated Manufacturing ERP Systems

Izabela Rojek[1] and Mieczysław Jagodziński[2]

[1] Kazimierz Wielki University in Bydgoszcz,
Institute of Mechanics and Applied Computer Science,
Chodkiewicza 30, 85-064 Bydgoszcz, Poland
izarojek@ukw.edu.pl
[2] Silesian University of Technology,
Institute of Automatic Control,
Akademicka 16, 44-100 Gliwice, Poland
mjagod@zeus.polsl.gliwice.pl

Abstract. The paper presents hybrid artificial intelligence system in constraint based scheduling of integrated manufacturing ERP systems. The system includes neural networks. The models were created by use simple neural networks (linear network - L, multi-layer network with error backpropagation - MLP and Radial Basis Function network - RBF) and hybrid neural networks in the form of: L - MLP network, L - RBF network, MLP-RBF network and L - MLP - RBF network. Neural networks as classification models were used to selection of tool for manufacturing operation. Next models as forecasting models were used to forecasting of tool use in different time intervals for manufacturing operation. These models were used at the stage of constraint bases scheduling and preventing standstill due to lack of tools, and special tools in particular. The created models were tested on real data from an enterprise.

Keywords: Neural network, classification, forecasting, intelligent support, constraint based scheduling.

1 Introduction

A task of management information systems is to support processes of enterprise management, understood as sequent process of decision making. Integrated information systems should be understood as "a module organised computer system, involved in all zones of enterprise activity (starting with marketing and supplying, manufacturing and its controlling, ending with distribution, sales, service, finances and human resources management)". Integrated enterprise management systems, including IFS Applications, coordinate flow and analysis of information regarding full product life cycle in framework of integrated supply chain SCM, it means from design to production planning, production with constraint based scheduling, controlling, supplying and service [1], [2].

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 229–240, 2012.

Contemporarily classic information systems are developed about new possibilities resulting from use of artificial intelligence methods [3], [4], [5] and estimating risk [6]. What is most important in intelligent systems is making conclusions [7]. The "intelligence" of the system is revealed through its ability to make decisions (via the conclusion making process), and also thorough its ability to learn and acquire knowledge. The intelligent systems, apart from classic information systems include neural networks, fuzzy systems, decision trees and genetic algorithms (evolution algorithms) [8], [9]. An intelligent system features the ability to acquire new knowledge, self-adapt, accept faulty or deficient data, and is creative at the same time. Systems of this type make it possible to monitor coefficients, to recognize potential threats early, to develop, and forecast possible action scenarios.

Research into available literature showed that researchers more and more often apply the methods of artificial intelligence in design of manufacturing processes. For example the selection of machining operations is aided [10], data mining methods are used in manufacturing [11], [12], [13] and the neural network models are created for the prediction of manufacturing parameters [14]. In the work [15] the authors introduced several examples of use of the expert systems in manufacturing process planning, production planning and control. The next application presents manufacturing knowledge model in the form of hierarchic decision networks [16].

Decision support systems can be used to manufacturing and constraint-based scheduling, too. During the constraint-based scheduling the selection of tools is a very important stage, with regard to large selection during design only, as and in situations of change of tools during process of the production. The experience of process engineer becomes necessary at this stage. With the aim of improving the selection of tools for manufacturing operation, the following was worked out:

- database, used for creating models and methods of intelligent aid,
- models of tool selection to manufacturing operations,
- forecasting models of tool use in different intervals of time,
- models of process engineer preference in tool selection.

With the aim of assuring the realization of a production plan, it is indispensable to create forecasting models of tools to use in different intervals of time. These models are used already at the design stage of a manufacturing process, so that the products are produced in proper time and the process does not come to standstill due to lack of tools, special tools in particular.

The paper includes a few sections. The first section relates introduction and state of art. The second section concerns description of constraint based scheduling. The next section includes neural networks as classification models for selection of tools for manufacturing operations, neural networks as forecasting models of tool use in different intervals of the time. Next selected tools are used in constraint based scheduling. The last sections apply to conclusion and references.

2 Description of Constraint Based Scheduling

Constraint-based scheduling (CBS) is a complicated task. It requires a detailed model of the problem and reacts to changes as they occur in the system while maintaining a feasible shop schedule. IFS/Constraint Based Scheduling is an add-on of an in-memory advanced Scheduling Server within IFS/Shop Order. This allows you to perform finite capacity scheduling in a resource and material constrained environment. With IFS/Constraint Based Scheduling, you can interact with the Scheduling Server in the same way as any other component in the IFS Applications framework, or use a powerful graphical user interface to work with the schedule interactively [17], [18].

IFS/Constraint-Based Scheduling solves the problem of when to complete a given set of jobs that represent work on a set of resources, e.g., machines, labor or tools, that can complete this work while attempting to meet some objective. The scheduling problem concerns three elements: to determine at a detailed level which work center, labor or tool resource should do what and when. This means that you must find an optimized assignment of resources to tasks, an optimized order in which the tasks are to be processed and an optimized time when the processing is to be done. There are different types of scheduling. Predictive scheduling makes an optimized schedule from a given set of orders. Reactive scheduling reacts to changes in the schedule and restores it to a useable condition. Interactive scheduling permits the direct manipulation of the schedule.Scheduling makes use of a production plan to develop a detailed schedule of the work to be done.

The Scheduling Server constructs schedules that are feasible with respect to the following constraints. Constraints can be divided into relaxable (e.g. capacity, material availability) and nonrelaxable (e.g. functional, availability, run time). A relaxable constraint is a usually a goal of the system that does not need to be met to provide a feasible schedule. A nonrelaxable constraint cannot be violated, because violating it will make the resulting schedule infeasible.

There are some key objectives to accomplish when constructing a schedule:

- minimize lead times,
- create a feasible schedule with respect to all nonrelaxable constraints,
- ensure that orders are not started too early, i.e., minimize WIP,
- minimize setups,
- minimize tardiness.

When constructing a schedule, many different steps are taken using different algorithms. First, material availability and resource assignment options are evaluated. Then, the earliest and latest possible start times for all operations are determined. Based upon a selected priority rule, these time values are used to sort the operations in the order in which they will be scheduled. The operations are then scheduled forward with the finite capacity constraints of the materials and resources.

3 Methodology of Creation of Models for Selection and Forecasting of Tools

3.1 Describe of Simple and Hybrid Neural Networks

Neural networks return continuous value at the output, hence they are excellent for estimating and classification [7]. Different types of neural networks were used for construction of the classification and prediction models.

In *multi-layer network with error backpropagation* (MLP), the signals flow from input to output. Multi-layer networks are formed with many layers of neurons. The input of each neuron from a given layer is linked with outputs of all neurons in the preceding layer [19]. The number of layers and neurons is random. The applied neuron model is of sigmoid type [8]. It consists of the sum element joined by input signals $x_1, x_2, ..., x_N$ in the form of an input vector $x = [x_1, x_2, ..., x_N]^T$ multiplied by weights $w_{i_1}, w_{i_2}, ..., w_{i_N}$ in the form of a weight vector of i-neuron $w_i = [w_{i_1}, w_{i_2}, ..., w_{i_N}]^T$ and the value w_{i_0}. The signal of sum element is marked u_i (Formula 1),

$$u_i = \sum (w_{ij} x_j + w_{i0}) \tag{1}$$

and the signal is given to non-linear activation function $f(u_i)$. It is a unipolar sigmoid function (Formula 2).

$$f_u(u_i) = \frac{1}{1 + exp(-\beta u_i)} \tag{2}$$

In order to assess the quality of knowledge acquired by the network during learning, three factors have been used: Root-Mean-Square Error (RMS), learning (training) tolerance and testing tolerance. The user can trace the above factors, and as a result determine the moment of termination of the learning process. RMS Error is the standard error, calculated as the sum of squared deviations of the actual and desired values, divided subsequently by the number of words (Formula 3).

$$RMS = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (T_i - O_i)^2} \tag{3}$$

where: n - number of instances, T_i - pattern value, O_i - real value.

Linear network - represented by a network with no hidden layers, however the neurons presented in the output layer are fully linear (i.e. there are neurons the collective stimulation of which is determined as a linear combination of input values and which have a linear activation function). During the operation of the network, the inputs are multiplied by weight matrix, which collectively forms a vector of output signals [9].

RBF - Radial Basis Function network usually has one hidden layer with radial neurons, each of which models a Gauss' surface of answers. Since these functions are strongly non-linear, one hidden layer is enough to model a function of any shape. Yet, for an RBF to form a successful model of any function, the network structure needs to dispose of many radial neurons. If there is a sufficient number of radial neurons, each important detail of a modeled function can have the needed radial neuron attached, which guarantees that the obtained solution shall genuinely reproduce the given function [9]. Radial networks consist of neurons, the activation of which functions are represented by formula 4:

$$x \to \varphi(\| x - c \|), x \in R^n \tag{4}$$

where $(\| \cdot \|)$ marks Euclidean norm.

Functions $\varphi(\| x - c \|)$ are called radial base functions. Their values change round centre c radially. Radial neuron is defined by its centre and a parameter defined as "ray". The neuron in output layer stands for the operation of the weighed sum of signals of output neurons in the hidden layer, it can be expressed with the formula 5:

$$y = \sum_i w_i \varphi_i = \sum_i w_i - \varphi(\| x - c \|) \tag{5}$$

The usage of linear networks, being analogous to linear regression function, constituted a clear starting point for further analysis with the use of other, more complex models. Three-layer perceptrons were used due to their universal possibilities, easy application and high probability of receiving positive results. The RBF network was used while looking for a better solution. The possible profits were short learning time and obtaining of good results. Hybrid neural networks were also developed for bettering model generalization.

In papers we meet the different applications of hybrid neural networks [20], [21], [22], [23]. They give better solutions than simple neural networks. Therefore the authors created classification and forecasting models in the form of hybrid neural networks, too.

Hybrid models constructed of different neural networks were drawn up (linear neural network, multi-layer neural network with error back propagation and Radial Basis Function network). The output for the hybrid model is composed of different network outputs. Two types of neural network sets were considered. In the first case, for classification purposes, the first prediction is obtained through voting (the winner takes it all) - the most represented value is the starting value for the set (a set with a winner). In the second case, the component networks are limited to a certain constraint. The complex output is formed on the level of output neurons. The sets of this kind average the outputs in all component networks (a set with an average). Sets equipped with an important tool against the overlearning of the network; these improve the generalization possibilities for the model. The quality of hybrid neural networks was assessed by cross-validation.

In next research, credibility of every neural network was considered on defining the final decision of hybrid neural network, similarly as in expert methods were taken into account credibility of every expert. The credibility of neural network in hybrid neural network was diverse across introduction of weights. These weights were used near voting and averaging. In this way some neural networks have larger or smaller influence on final result. The earlier research showed diverse accuracy of every network. RBF network was the best under regard classifying, MLP network a bit worse and linear network the least exact. Therefore differentiation of their credibility was executed. RBF network received the highest weight (weight = 1), and linear network the lowest (weight = 0,1). Weights were chosen in experimental way (figure 1).

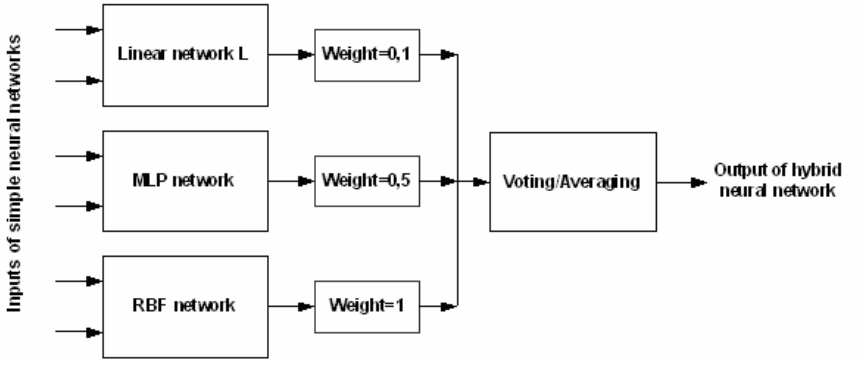


Fig. 1. Hybrid neural networks with differential credibility

With a view to selection of tools for manufacturing processes in a comprehensive way, classification and prediction models were developed. The following computer models developed:

- selection of tools - classification models,
- forecasting of tool use in different intervals of time - prediction models.

The models are compared for obtaining better classification and prediction. The models were tested on data from the real enterprise. Basing on the classification and prediction models, intelligent support system allows to create scenarios for selection of tools for manufacturing operations. That is why the created models make the manufacturing processes better.

3.2 Neural Networks for Selection of Tools for Constraint Based Scheduling

Data Preparation. We suppose that representative number of cases of tools selection is located in database. Correctness of acquired knowledge greatly depends on data examples on the basis of which methods of knowledge acquisition obtained this knowledge. Learning and testing files were prepared with the

aim of knowledge acquisition aiding tool selection for manufacturing operation. Examples of tool selection divided thematically into learning and testing files, separately for turning, milling and grinding. Learning files include about 900 examples, whereas testing files about 200. Input attributes are nominal, ordinal and numeric type. "Data purification" process was carried out. An accurate data profiling, data parsing, data verification (on level of a field, a row, a table) and data standardization was carried out. Data duplication was removed, too. The examples included in files are real tool selections carried out during design of manufacturing processes in the enterprise.

For example in the case of *milling*, input data includes: the kind of milling (e.g. roughing), type of machining surface (e.g. surface), stock symbol (e.g. 1,053), type of stock (e.g. soft), milling tool structure (e.g. plating), demanded surface roughness (e.g. 40), milling tool structure (e.g. inserted-tooth cutter), kind of milling tool clamping (e.g. arbor), dimension (e.g. 160), tooth number (e.g. 10), total length of milling tool (e.g. 300). Output data is milling tool symbol (e.g. hR 257 .1-160).

Experiments with Classification Models. Experiments were carried out for selected manufacturing operations: turning, milling and grinding. For each operation, intelligent models were drawn using all types of neural networks: L, MLP, RBF, L-MLP, L-RBF, MLP-RBF, L-MLP-RBF network. Analysis and comparison of classification models for selection of tools were carried out. The paper was illustrated with selected models.

In order to assess the quality of knowledge acquired by the network during learning, three factors have been used: Root-Mean-Square Error (RMS), learning (training) tolerance and testing tolerance. The user can trace the above factors, and as a result determine the moment of termination of the learning process. RMS Error is the standard error, calculated as the sum of squared deviations of the actual and desired values, divided subsequently by the number of words. All these models classify the tools for manufacturing operations.

Having analyzed various classification models for selection of tools for manufacturing operations, it was the MLP-RBF network model which proved the most effective (figure 2).

The difference in correct classifying of the simple neural networks, hybrid neural networks with equal credibility and hybrid neural networks with differential credibility was showed in figure 2. Arithmetical average of values of proper classification of RBF, MLP and L networks was counted for comparison. Classification quality of hybrid neural networks submitted improvement after the introduction of differentiation of networks in hybrid neural network. The MLP-RBF model with differential credibility was the best.

Using neural networks as classification models, application having aid process engineer in selection of tools for manufacturing operation was implemented. The application of tool selection in dialog form queries process engineer about input attributes and answers in the form of tool symbol.

Simple neural network	Proper classification (%)		
L	64,14		
MLP	97,59		
RBF	98,00		
Hybrid neural network	Average of proper classification of simple neural networks (%)	Proper classification of hybrid neural network about equal credibility of simple neural networks (%)	Proper classification of hybrid neural network about differential credibility of simple neural networks (%)
L-MLP	80,87	92,08	93,23
MLP-RBF	**97,80**	**98,02**	**98,80**
L-RBF	81,07	96,86	97,86
L-MLP-RBF	86,58	95,57	98,01

Fig. 2. Comparison of values of proper classifications of selection models [4]

3.3 Neural Networks as Forecasting Models of Tool Use in Different Intervals of the Time

Nowadays, the use of tools in enterprises is controlled on the general level. The number and condition of tools is checked in tool-houses according to a predefined time schedule.

However the worked out models will permit to forecast requirements for particular tools in different temporary intervals and react more quickly to the lacks of tools in an enterprise, which will prevent the standstills of production and also the increase of production costs.

Data Preparation. The forecasting models permitted to forecast which tools were used for which manufacturing operations most often and in what time. This type of models permits to prepare the tool to realization of manufacturing operations in advance and to react to lack of tools, e.g. through generating purchase orders for the missing tools. Learning and testing file were prepared to teach prediction models in form of neural networks [5]. The learning set included about 300 instances and the testing file about 60.

Knowing the model of the facility, the reaction to various input violations should be analyzed. It is interesting to define the future state of the facility for the time $t+n$, where n is the prediction horizon, t contains the input changes history up to the present. The prediction horizon n=1 marks e.g. 1 day. In order to construct time sequences which are later used in forecasting model, the values of time of tool use before the moment t (t-1, t-2, ..., t-7) and after the moment t

(t+1) were added. The experiments were also conducted for values from before t (t - 30) and after t (t + 1) marking prognosis on the month, which confirmed the correctness of models [5].

Assessment of Forecasting Models. The assessment of the quality of neural prediction models may be done by comparing the graphs: the real and the forecast. It is a very common form of presenting research results. Yet the quantitative methods for assessment of neural models are the ones which allow to formulate more objective conclusions. The assessment of neural models is generally carried out in two phases. In the first phase, after having constructed the taught networks, assessment is done through the so-called regressive statistics. The quotient of standard deviation and correlation of real and forecasted values are the most important for assessment of neural models [24]. The first parameter for the constructed models should assume the values of 0,1−0,2.

The deviation quotient with a value close to zero testifies to a good value of a given model. If this is greater than one (or close to one) then the designed model may be rejected. It is difficult to clearly define the correctness of the model if the deviation quotient falls within the range of 0,3−0,7. The quality of the model must eventually be obtained form the obtained ex post errors, or their acceptability in the particular case. The correlation between the real and forecasted values assumes values form 0−1. It is best when close to one (the closer, the better). The quotient of standard deviation = quotient error deviation / standard deviation of real data.

Correlation is practically standard Pearson correlation coefficient r (Formula 6).

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{6}$$

where: $\overline{x}, \overline{y}$ - average values, x_i, y_i - values of random tests.

Coefficient r is appointed for real and forecasted values of working time of tool. It is parameter of assessment.

In the second phase, after having taught the network, the forecasting process is carried out. Except the real values, the prediction of output variable for the model is obtained. It allows to determine the ex post errors. We should dispose value double: real and appointed for help of given model to calculation of *ex post* errors. We can include sum of squares of residuals, mean squared error (MSE) and root mean squared error (RMS) to the most often the applied measures of quality of neural models. The above defined quality measurements of the neural model prediction errors were given in the experiment phase. The value of the RMS errors of learning and testing were given with the drawn up models. The model with the smallest RMS error and the coefficient of number of patterns behind tolerance was chosen. The results used from forecasting models were also compared with real values of tool use in forecasting intervals of time.

Neural networks generalized well, because answers given by them were comprised in the range of error for testing file.

Experiments with Forecasting Models. Experiments were carried out for selected manufacturing operations: turning, milling and grinding. For each operation, intelligent models were drawn using all types of neural networks: L, MLP, RBF, L-MLP, L-RBF, MLP-RBF, L-MLP-RBF network. Analysis and comparison of forecasting models for tool use in different intervals of times were carried out. The paper was illustrated with selected models. The models are equipped with inputs (tool use in time: t-7, t-6, t-5, t-4, t-3, t-2, t-1, t) and one output (tool use in t+1) and one hidden layer containing (5, 10 or 15 neurons for MLP network and 7, 15 or 20 neurons for RBF network). These structures were learnt with different conditions of ending the process, i.e. the end after reaching the number of periods equal to 1000, 10000 or 100000. To every combination, a RMS error was compared. Having analyzed various forecasting models, it was the MLP-RBF network model about differential credibility which proved the most effective.

The assessment of the quality of neural prediction models may be done by comparing the graphs: the real and the forecast. The assessment of neural models is generally carried out in two phases. In the first phase, after having constructed the taught networks, assessment is done through the so-called regressive statistics. Figure 3 shows regressive statistics for weekly prediction models.

Simple and hybrid neural network	Quotient of standard deviation	Correlation
L	0.42828	0.70853
MLP	0.21733	0.73902
RBF	0.18783	0.84072
L-MLP	0.26476	0.76016
L-RBF	0.25343	0.91957
MLP-RBF	**0.18712**	**0.96081**
L-MLP-RBF	0.23384	0.91069

Fig. 3. Regression statistics of prediction models for week's prognosis [5]

On the basis of classification and prediction models, decision rules in the intelligent decision support system of selection and forecasting of tools for constraint based scheduling were created. Using neural networks as classification and forecasting models, application having aid process engineer in selection of tools for manufacturing operation was implemented. The application of tool selection, in dialog form queries process engineer about input attributes and answers in the form of tool symbol and the time, when tool will be used. Intelligent system, on the base of rules, models, and methods support of selection of tools.

4 Conclusion

IFS Applications system assures complete aided complex technical environmental in enterprise. The modules were described in paper; make up part of IFS Applications system, which support all business processes in an enterprise. IFS Applications system makes possible full support for complex technical environment in this flow coordination of information concerning product life cycle. Application of IFS system in an enterprise guarantees economic growth of enterprises and quick adaptation to changeable market conditions that allows gaining considerably competitive advantage.

The application of neural networks in aid of constraint based scheduling, in selection of tool and prediction of tool use in determined temporary intervals, introduced a new quality to the constraint based scheduling systems and it can stand as a foundation of algorithmization of the new so-called "intelligent" systems. Using neural networks as classification and forecasting models, application having aid process engineer in selection of tools for manufacturing operation was implemented. The application of tool selection, in dialog form queries process engineer about input attributes and answers in the form of tool symbol and the time, when tool will be used.

Application of artificial intelligence methods enables creation aided system, which collects knowledge automatically and has adaptation skills. It is particularly important when working out a system for complex real systems, in which continuous changes follow and a single process depends on another, many factors depend on one another and every change triggers next changes.

References

1. Thompson, K.N.: Product Life Cycles: Theoretical & Practical Issues. University of North Texas, John Wiley & Sons (2001)
2. Rojek-Mikołajczak, I., Jagodziński, M.: Product life cycle in integrated system of enterprise management. In: Virtual Design and Automation, pp. 269–275. Poznan University of Technology, Poznań (2005)
3. Rojek, I.: Neural Networks as Performance Improvement Models in Intelligent CAPP Systems. In: Performance Evaluation Models of Manufacturing Systems, Control and Cybernetics, Warsaw, vol. 39(1), pp. 55–68 (2010)
4. Rojek, I.: Support of decision making processes and control in systems with different scale of complexity using artificial intelligence methods. Publishing House of Kazimierz Wielki University, Bydgoszcz (2010) (in Polish)
5. Rojek, I.: Forecasting Models of Tool Use in Different Intervals of Time. Management and Production Engineering Review (MPER) 1(1), 31–39 (2010)
6. Burduk, A.: An attempt to adapt serial reliability structures for the needs of analyses and assessments of the risk in production systems. Maintenance and Reliability (3), 85–96 (2010)
7. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall (2009)
8. Larose, D.T.: Discovering Knowledge in Data: An Introduction to Data Mining. John Wiley & Sons (2005)

9. Rutkowski, L.: Computational intelligence, methods and techniques. Springer, Heidelberg (2008)
10. Deb, S., Ghosh, K., Paul, S.: A neural network based methodology for machining operations selection in Computer Aided Process Planning for rotationally symmetrical parts. Intelligent Manufacturing 17(5), 557–569 (2006)
11. Kusiak, A., Smith, M.: Data mining in design of products and production systems. IFAC Annals Reviews in Control 31(1), 147–156 (2007)
12. Rokach, L., Maimon, O.: Data mining for improving the quality of manufacturing: a feature set decomposition approach. Intelligent Manufacturing 17(3), 285–299 (2006)
13. Wang, K.: Applying data mining to manufacturing: The nature and implications. Intelligent Manufacturing 18(4), 487–495 (2007)
14. Markopoulos, A.P., Mandakos, D.E., Vaxevanidis, N.: Artificial neural networks models fort he prediction of surface roughness in electrical discharge machining. Intelligent Manufacturing 19(3), 283–292 (2008)
15. Knosala, R. (ed.): Applications of artificial intelligence methods in production engineering. WNT, Warsaw (2002) (in Polish)
16. Duda, J., Habel, J., Pobożniak, J.: Use of Manufacturing Knowledge for Process Planning in Distributed Environment. In: Virtual Design and Automation, pp. 187–194. Publishing House of Poznan University of Technology, Poznań (2005)
17. Krystek, J., Jagodziński, M., Rochowiak, T.: Constraint-based scheduling in IFS Applications. Polish Academy of Sciences, vol. 49, pp. 161–173. System Research Institute, Warsaw (2006) (in Polish)
18. Jagodziński, M., Kołodziej, Z., Krystek, J., Kusek, M.: Production scheduling in modern MRP/ERP systems, Gliwice. Scientific Papers of Silesian University of Technology, Automation, vol. 135, pp. 65–76 (2002) (in Polish)
19. Haykin, S.: Neural Networks: A Comprehensive Foundation. Pearson Prentice Hall (2001)
20. Caciotta, M., Giarnetti, S., Leccese, F.: Hybrid Neural Network System for Electric Load Forecasting of Tele-comunication Station. In: Fundamental and Applied Metrology, pp. 657–661. House of Poznan University of Technology, Lisbon (2009)
21. Chen, H., Grant-Muller, S., Mussone, L., Montgomery, F.: A Study of Hybrid Neural Network Approaches and the Effects of Missing Data on Traffic Forecasting. Journal Neural Computing and Applications 10(3), 277–286 (2001)
22. Smaoui, N.: A Hybrid Neural Network Model for the Dynamics of the Kuramoto-Sivashinsky Equation. In: Mathematical Problems in Engineering, pp. 305–321. Hindawi Publishing Corporation, Hindawi (2004)
23. Tsai, C.-F., McGarry, K., Tait, J.: Image Classification Using Hybrid Neural Networks. In: ACM Conference on Research and Development in Information Retrieval, New York, pp. 431–432 (2003)
24. Tadeusiewicz, R., Lula, P.: Statistica Neural Networks 4.0 PL: Introduction to neural networks. StatSoft Poland, Cracow (2001) (in Polish)

Intelligent Data Processing in Recycling of Household Appliances

Edward Chlebus, Kamil Krot, Michał Kuliberda, and Bolesław Jodkowski

Institute of Production Engineering and Automation, Wrocław University of Technology
Lukasiewicza st 50-371 Wroclaw, Poland
{Edward.Chlebus,Kamil.Krot,Michal.Kuliberda,
Boleslaw.Jodkowski}@pwr.wroc.pl

Abstract. This paper presents processes of numerical data processing and objects digitalizing in recycling of household appliances. General description of technological line designed for refrigerators recycling. Processes of refrigerator housing scanning and scanned data processing are presented. Algorithm of scanning process, data filtration and key parameters getting are described in details. Moreover, database designed for storing scanned data is presented.

Keywords: Recycling, vision systems, data processing.

1 Introduction

Recycling of household appliances, in particular refrigerators, became a very important issue strictly connected with the protection of the environment. The most dangerous chemical compound contained in old refrigerators is dichlorodifluoromethane CFC-12 (R-12). CFC-12 belongs to the family of chemical compounds called freons and influences the environment. It is believed that CFC-12 causes lowering of the ozone concentration in Earth's stratosphere which in turn causes ozone depletion. To stop this process the use of freons in new products became prohibited.

According to Directive of the European Parliament on Waste Electrical and Electronic Equipment (WEEE) refrigerators recycling may be performed only in specializes institutions in accordance with the local law. Currently the process of refrigerator recycling is focused on removing dangerous components. Next oil from compressor circuit is removed and the compressor is treated as a scrap and is not further processed. On the last stage of a common refrigerator recycling line housings are milled. W Institute of Production Engineering and Automation of Wroclaw University of Technology a co-financed by UE funds project regarding refrigerator recycling is ran. It is entitled: „Technological line for recycling of household goods with the use of laser processing" WND-POIG.01.03.01-02-046/08 within the framework of the Programme Innovative Economy 2007-2013. Scientific works are focused on development of refrigerator recycling line employing innovative solutions for recycling process. One of innovative solutions is vision identification of refrigerator housings employing laser scanning and digital models processing.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 241–249, 2012.

2 Refrigerator Recycling

The developed for refrigerator recycling technological line consists of following key stations (fig. 1):

- refrigerator storage,
- manual disassembly station (removing of coolant, shelves, drawer, condenser, compressor),
- vision identification of housing (setting position, processing of data, scanning, storing in the database),
- laser machining of housing (laser cutting of boards from refrigerator housings basing on scanned data,
- shredding and sorting residual parts of refrigerator housings (magnetic separators, eddy currents),
- processing of boards cut from,
- recycling products storage.

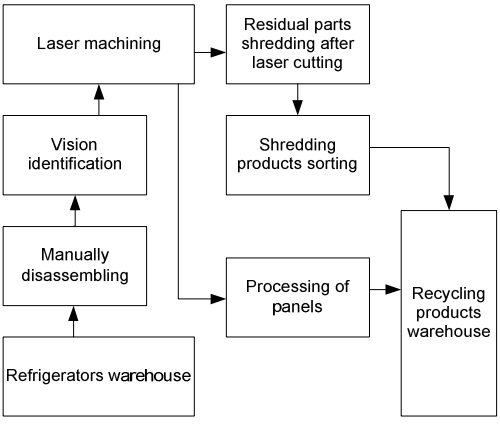


Fig. 1. Schematic of refrigerator recycling line

This paper presents an innovative approach to the recycling process which is focused on the restoration of original functions of the processed device. Cabinet shredding followed by separation processes which is the most common way to the refrigerator recycling is applied only to residual parts of refrigerator housing [1]. Presented approach regards only to refrigerator housing which makes the biggest part of a refrigerator volume. The structure of refrigerator housing can be classified as a sandwich composite. Its main functions are: ensuring of thermal insulation from the environment, making the device rigid enough and allowing installation of necessary electrical equipment inside of the composite. Typical sandwich composites used to build refrigerator housings consist of following components: external shell made out of steel, insulating layer of foamed polyurethane and an inner shell made out of plastic [3]. The structure of this composite is shown in fig. 1. Insulating properties of

Fig. 2. Section of the sandwich composite refrigerator housing: 1 – steel plate, 2 – polyurethane foam, 3 - plastic

the sandwich composite can be used for other purposes, e.g. as an thermal insulation material in house building industry.

To use selected parts of refrigerator housing for such a purpose repetitive shapes must be obtained. Authors use laser machining which allows cutting out any shape from the refrigerator housing. A laser beam is driven by a robot [4]. A robot control program and parameters of the laser machining depend on the parameters of currently processed refrigerator housing. To provide the recycling line with such functionality employing of a vision identification system and intelligent data processing become necessary. These data will be used for successfully complete the laser cutting process. All recycling processes can be simulated and after it optimal parameters must be applied to real recycling line [5].

3 Scanning of a Refrigerator Housing

Vision identification of refrigerator housing allows obtaining spatial set of point describing housing geometry. Transformations of this set leads to detection of housing wall thickness and location of a baffle between cooler and freezer.

Authors used the LMS 400 Laser Measurement System for scanning refrigerator housing geometry. This device is a laser range finder which returns data in the polar coordinate system. Single measurement provide us with a list of points described by a distance from the laser source and the angle. It is also possible to measure the value of remission (the possibility to reflect light back) of the measured surface. However the device scans parts of objects laying in one planar surface – returns points laying in two dimensional coordinate system, moving the scanner along axis perpendicular to this surface allows us to scan three dimensional objects.

In one single measurements the LMS400 returns not only coordinates of points. The message frame include many additional information which are repeated in every measurement cycle. These additional data describe parameters like: scanner settings, scanning parameters, digital inputs states and encoder state. Points are returned in the form of list of distances, the angular step I constant and is returned once for every measured line. Combination of points coordinates in polar coordinate system with the encoder state assuming linear movement of the scanner gives data describing three

dimensional model of refrigerator housing. To make processing of these data easier some additional information are omitted and point coordinates and encoder states are transformed in the way described in table 1.

Table 1. Transformation of point cloud receiver from LMS 400

0	(encPos x) * scannMoveFactor	startAng + 0*(angStepWidth)	(dist 0) * (distScalling)
1	(encPos x) * scannMoveFactor	startAng + 1*(angStepWidth)	(dist 1) * (distScalling)
2	(encPos x) * scannMoveFactor	startAng + 2*(angStepWidth)	(dist 2) * (distScalling)
…	…	…	…
…	…	…	…
n	(encPos x) * scannMoveFactor	startAng + n*(angStepWidth)	(dist n) * (distScalling)

The term *scannMoveFactor* describes the dependency between encoder steps and scanner movements. The value x close to the *encPos* parameter represents the number of measured line. The final data structure includes all measured lines, three columns and number of rows equal to equation (1).

$$n * \text{number of lines} \tag{1}$$

This structure include the whole scanned point cloud which is ready to further processing. Scanning station is presented in fig. 3. Figure 4 shows a photo taken during scanning of refrigerator housing.

Having the hole housing measured allows visualization of the measured refrigerator in the form of 3D charts – fig. 5.

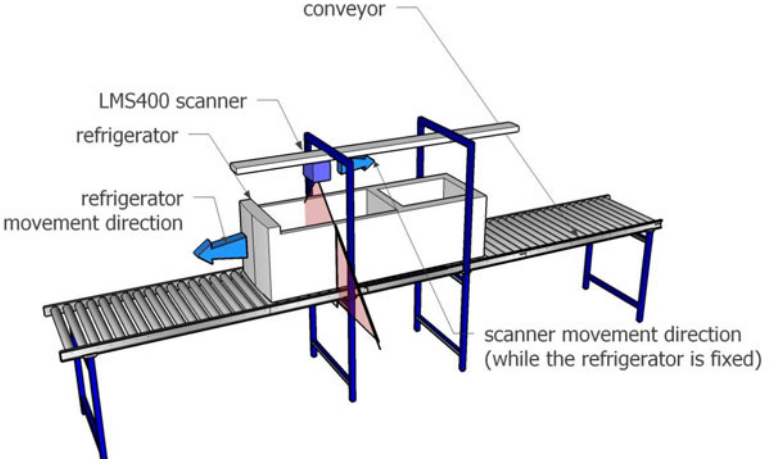


Fig. 3. A part of technological line responsible for scanning of refrigerator housing

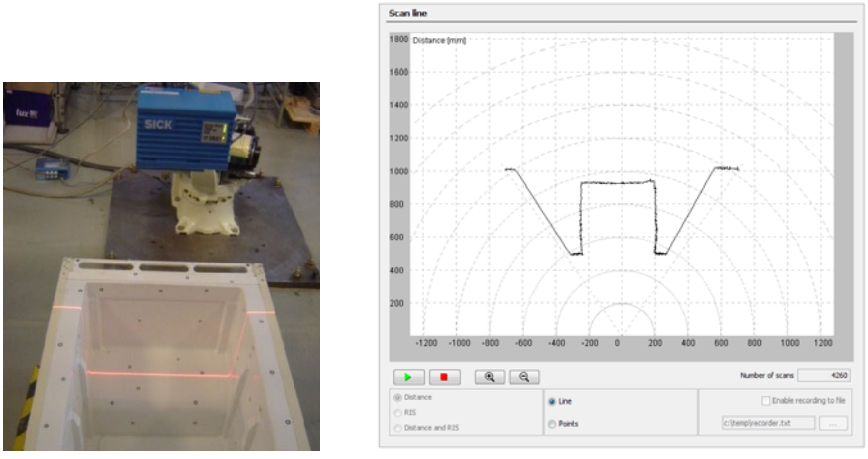


Fig. 4. Refrigerator housing scanning on the left, currently measured part of housing on the right

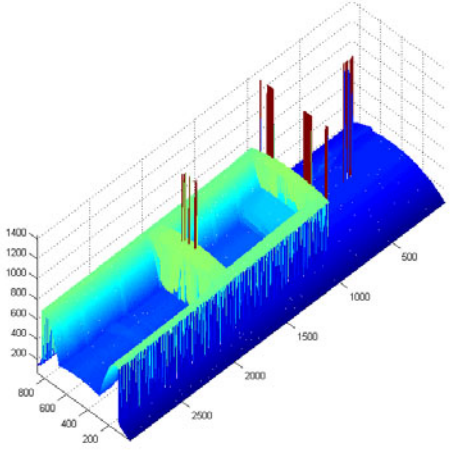


Fig. 5. Results of measurements: light green signifies high Z values, dark blue – low values

Detailed processing of these data will be described in the next chapter.

4 Data Processing

Obtained point cloud may consist of up to 254000 points (up to 1000 lines and 254 point in every line). Making more precise measurements can make this number a few times bigger. The more precise measurements the bigger point cloud. Some of obligatory operations on the row point cloud include removing disturbances in the form of single points located in places where was not any part of housing geometry. Such operations require analysis of surround of points which require calculating

distance between points, such a process would require calculating of this distance number of times equal to 6,45 * 10^{10}.

To reduce the number of obligatory calculations authors decided to transform the point cloud to three dimensional matrix having constant resolution. Values of particular fields are increased by one in case when a point is located in the piece of space constrained by matrix filed dimension. This process is described below:

- the 3D matrix has dimensions *A* x *B* x *C* and has rectangular shape,
- matrix entries are identified using indices *indA*, *indB*, *indC*,
- matrix dimensions are defined according to equations (2)-(4).

$$dimA = \max_m p_{m_x} - \min_m p_{m_x} \quad (2)$$

$$dimB = \max_m p_{m_y} - \min_m p_{m_y} \quad (3)$$

$$dimC = \max_m p_{m_z} - \min_m p_{m_z} \quad (4)$$

where:

p_{m_i} represents i-coordinate ($i = x, y, z$) of point number m belonging to the point cloud. In practice matrix covers the most distant points of the point cloud,

- every matrix entry covers a rectangular piece of space, which dimensions are defined by equation (5).

$$size_n = \left(\frac{dimA}{A}, \frac{dimB}{B}, \frac{dimC}{C}\right) (5)$$

where *n* represents the number of matrix entry,

- coordinates of a starting point of every matrix entry are defined by equation (6).

$$pos_n = \left(\frac{dimA}{A} \cdot indA, \frac{dimB}{B} \cdot indB, \frac{dimC}{C} \cdot indC\right) (6)$$

where n represents the number of matrix entry,

- indices of a matrix entry where current point belong to are calculated using following equation (7).

$$indA_{zapis} = \left\lfloor \frac{punkt_{m_x}}{dimA} \right\rfloor (7)$$

remaining indices are calculated in a similar way.

Figure 6 represents the process of the transformation of the point cloud into the matrix. To make it simpler to demonstrate 2D has been used.

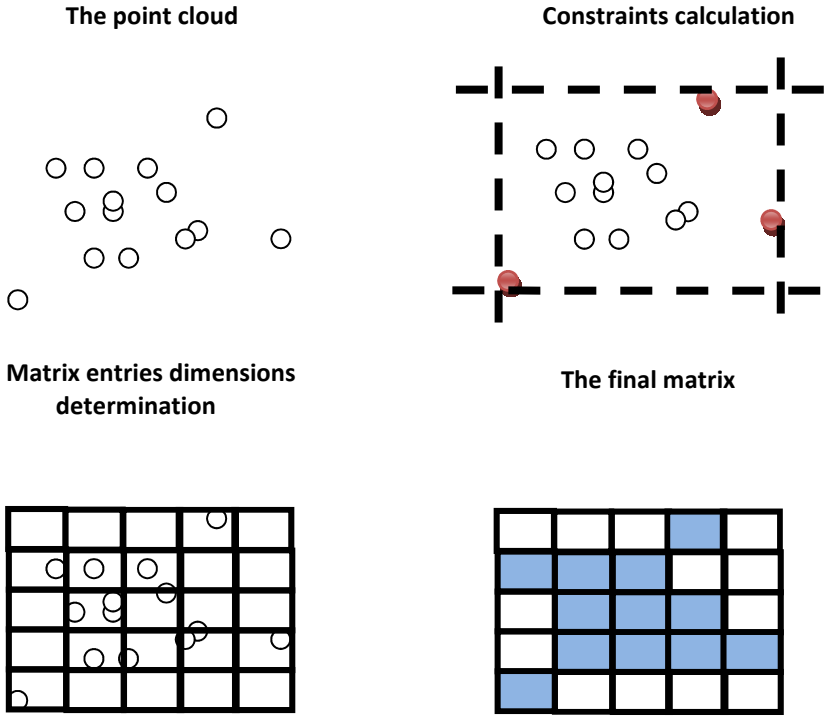


Fig. 6. The process of transformation of the point cloud into the form of matrix (simplified 2D version)

The algorithm of transformation of the point cloud into 3D matrix is presented on fig. 7. Presented algorithm bases on Cartesian coordinate system. To make it easier to implement it is necessary to convert the data structure from table 1 to the data structure described by formulas (8) - (11).

$$p_{m_z} = EP_i \cdot WPL (8)$$

$$\propto = n \cdot ASW - \frac{NMV \cdot ASW}{2} (9)$$

$$p_{m_x} = \mathrm{Dist_n} \cdot \mathrm{DS} \cdot \sin(\alpha) \ (10)$$

$$p_{m_y} = \mathrm{Dist_n} \cdot \mathrm{DS} \cdot \cos(\alpha) (11)$$

where:

EP_i – encoder position for an i-measurement
WPL – real scanner movement factor,
ASW – angular step width,
NMV – number of points in the measured line,

DS – distance scaling factor,
Dist$_n$ - *Distance_n* – the distance between n-point and laser source for the current line,
n – point number.

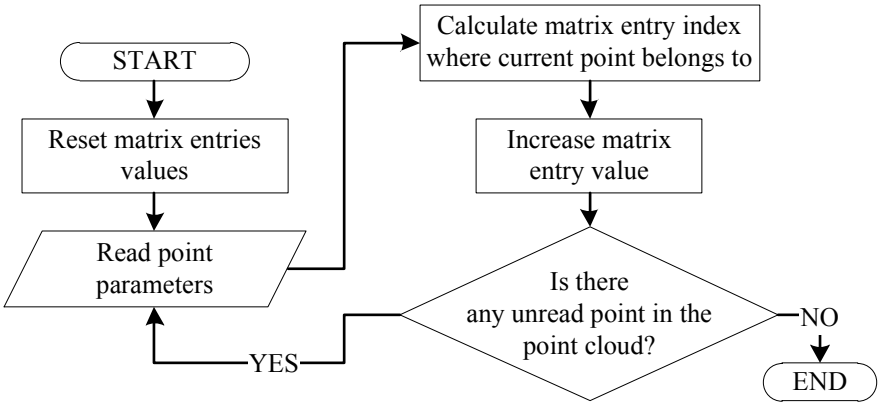


Fig. 7. Flowchart of the algorithm transforming the point cloud into the 3D matrix

5 Summary

In this article authors described the most important part of an innovative refrigerators recycling process. The method is focused on preserving and making use of original properties of refrigerator housings i.e. heat insulation. Instead of crushing a worn refrigerator, in the developed approach the biggest possible blocks are cut from refrigerator housings. These blocks can be then used as a heat insulation material. To make this work it is necessary to recognize locations of these parts of housings which can be removed. Authors developed vision identification station for this purpose. Scanned data representing currently processed housing are sent to the computer where are transformed in order to identify flat parts of housings which are then removed using laser machining. Presented approach significantly reduces the risk of leaking dangerous chemical compound to the environment.

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. Shah, J., Mäntylä, M.: Parametric and feature-based CAD/CAM. John Wiley & Sons, New York (1995)

4. Kuliberda, M.: Opracowanie wybranych modułów opartego na wiedzy, generacyjnego systemu CAPP. Instytut Technologii Maszyn i Automatyzacji. Instytut Technologii Maszyn i Automatyzacji Politechniki Wrocławskiej, Rozprawa doktorska (2009)
5. Chlebus, E., Krot, K., Kuliberda, M.: Dekompozycja modeli CAD 3D w planowaniu procesów technologicznych. In: XV Konferencja nt. Metody i środki projektowania wspomaganego komputerowo, pp. 55–62 (2005)
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. Krot, K., Kuliberda, M.: Identyfikacja technologicznych obiektów elementarnych części maszyn w modelach geometrycznych CAD 3D w reprezentacji brzegowej. Inżynieria Produkcji Wiedza - Wizja - Programy ramowe, strony, 169–176 (2006)
9. Marchetta, M.G., Forradellas, R.Q.: An artificial intelligence planning approach to manufacturing feature recognition. Computer-Aided Design, 248–256 (2010)
10. Krot, K.: Opracowanie systemu wspomagającego planowanie procesów obróbkowych metodą obiektów elementarnych. Instytut Technologii Maszyn i Automatyzacji Politechniki Wrocławskiej, Rozprawa doktorska (2005)

Assessment of Risk in a Production System with the Use of the FMEA Analysis and Linguistic Variables

Anna Burduk

Wrocław University of Technology, 27 Wybrzeże Wyspiańskiego St,
50-370 Wrocław, Poland
Anna.Burduk@pwr.wroc.pl

Abstract. The paper describes a method for analysing and assessing the risk in production systems. A process of ore transportation process with the use of a belt conveyor was used as an example. Ishikawa diagram was used to identify the risk factors in the cause and effect analysis. In order to determine the extent of the impact of individual risk factors on the selected area of the production system, the FMEA analysis was used. When determining the values of the parameters needed for calculating the Risk Priority Number (RPN), defuzzified values of appropriate linguistic variables were used. The effect of the work is a reduction of the risk level in the analysed production system as well as the information about risk factors obtained on the basis of verbal communications from production workers.

Keywords: Risk, production system, transportation of excavated material, FMEA analysis.

1 Introduction

Nowadays, companies focus their attention primarily on operational and organizational problems. Risk, which is a natural and common phenomenon in enterprises, is a fundamental issue here. Elimination of risk is impossible, because it affects every decision.

Business activity is characterized by uncertainty. This condition is caused by many factors, which include, inter alia, a large number of elements making up a production system and the dynamics of the system. A measure of the uncertainty (ignorance) of the state is the average entropy of the state of the H(X) object defined by the equation (1):

$$H(X) = -\sum_i p(x_i) \log_a p(x_i) \tag{1}$$

where: X_i-th state of the x object, $p(x_i)$ - probability of the occurrence of the x state, a- radix. Usually it is assumed that $a = 2$.

Planning and decision-making processes in contemporary companies generally use deterministic methods, without taking into account the conditions of uncertainty [2]. This increases the risk, because there is no information about the possible occurrence

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 250–258, 2012.

of threats and the resulting effects. To mitigate the risk and increase the probability of taking correct decisions, actions should be taken in order to identify the area of risk, its extent and the impact on the operations in the organization, as well to search for measures for eliminating the risk. The awareness of the omnipresence of various types of risk raises the need to identify it in terms of the place of its occurrence and the strength of its impact on the company.

In the case of mining processes it is particularly difficult to assess the impact of risk factors on a production system. This is caused by specific conditions, in which the processes run, as well as by the provisions of the mining law. The information about risk factors often comes from production workers and has a linguistic value determined without data from technical measuring instruments. In the further part of this study, a linguistic variable was used to assess the risk of a failure of a belt conveyor.

2 Identification and Assessment of Risk in Production Systems

In order to reduce the level of risk in a production system, a series of actions must be taken. The first of them is the risk identification, which determines the threats that might occur during realization of company's goals. Due to a potential possibility that many risk factors may occur, it is important to find the source risk, which is the key cause of the problems. During the identification, it is important to search for the answers to the following questions: in which area of the production system the risk occurs and which area is affected by the highest risk.

The next step in reducing the risk level is measuring the risk and determining the extent of the impact on the production system. Failure Mode and Effect Analysis (FMEA) is one of the methods which allow determining the extent of risk in the designated area of a production process or in a product, as well as the resulting effects. Thanks to this, corrective actions aiming at mitigation of the risk can be found subsequently [6]. "*One of the key factors in proper implementation of the FMEA program is to act before an event occurs and not to gain experience after the event. In order to obtain the best results, FMEA should be performed before a particular type of construction or process defect is "designed*" for a given product." [3].

2.1 Determination of the Risk Priority Number (RPN) in the FMEA Method

When assessing the risk in a production process with the use of the FMEA method, the first step is to detail the operations in the process, then to identify the risk factors present in the process, determine the effects caused by their presence, and to find possible causes. The next step in the analysis is to assign numerical values to the following parameters shown in Table 1.

Risk Priority Number (RPN), i.e. the extent of the risk, is calculated for each of the selected areas of the production system using the formula [5]:

$$RPN = (Z) \times (P) \times (T) \tag{2}$$

Table 1. Characteristics of the parameters used in the FMEA method for determining RPN

Parameter symbol	Parameter name	Description
Z	degree of threat	It determines the extent of the effects which arise as a result of the occurrence of a defect during a production process and during the use of a product.
P	probability	The probability of the occurrence of a defect
T	detection rate	It determines the probability that a potential defect or its cause will be detected later

This obtained value allows assessing the estimated risk and is used as a point of reference in relation to the corrective actions taken. The value of RPN may be in the range between 1 and 1000. So a high value of RPN corresponds to a high risk in the process. If the RPN value is high, efforts should be taken to mitigate the risk using corrective actions [3]. The corrective actions shall be taken first in the areas with the highest RPN level.

Fig. 1 shows 4 areas representing an area of high losses and risk. These areas are presented together with the parameters described above.

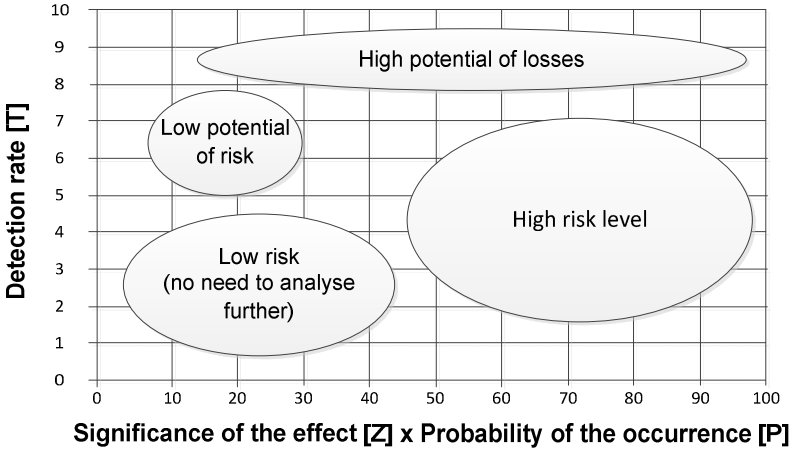


Fig. 1. The results of the RPN analysis depending on values of the parameter

Determination of a general limit for a high RPN value is not easy. Each FMEA analysis is unique and the risk estimation in this method cannot be compared with other analyses. This is caused by some sort of subjectivity, the dependence during the assessment, and the decisions made by the person performing the analysis. Therefore for each FMEA analysis a system of criteria should be developed and it should be determined from which values of RPN the corrective actions should be taken [5].

3 Determination of Risk in the Process of Haulage of Excavated Material by Belt Conveyors, Using Linguistic Variables

In the "Rudna" Mining Enterprise, located in the Lubin Copper Basin, haulage of excavated material is carried out with the use of belt conveyors. The belt conveyor transport system consists of 65 conveyors with a total length of approx. 46 km. The conveyors are connected with holding tanks in nodal points.

Belt conveyors are mechanical means of transport with a limited range and continuous movement. Typically they are used for conveying bulk materials. Material is transported on a specific route limited by the distance between the loading and unloading stations. Depending on the construction, material can be transported along a straight line or a curve, at any angle. Belt conveyors are characterized by simple construction, high reliability and safety. More and more often they are used also for transporting people.

The main components of a belt conveyor ate shown in Fig. 2. These parts form a serial structure, which means that the correct operation of each subassembly has a direct effect on the functioning of the conveyor [1].

Fig. 2. Diagram showing the reliability structure of a belt conveyor

The problem of failures of belt conveyors was subjected to an analysis. This is a very important issue in respect of transportation of excavated material in a mine, because failures lead to unplanned downtimes and thus to stopping the haulage of excavated material for several shifts. On the other hand, the information about a failure may come from production workers only, which results from the conditions occurring in a mine, the length of the transport system and provisions of the mining law. Information about a failure was verbal and depended on individual impressions of workers.

Fig. 3 presents a cause and effect analysis of belt conveyor failures in the form of Ishikawa diagram. A failure of a belt conveyor, i.e. an interruption in its operation, was assumed as an effect. Risk factors were divided into the main factors and presented in boxes. Then the causes of their occurrence were analysed and were decomposed to the third level on this basis.

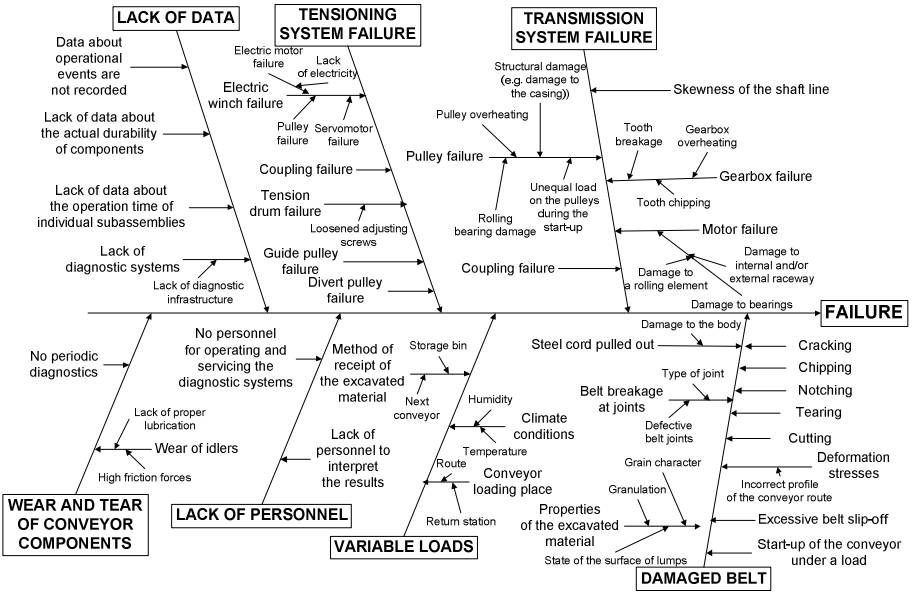


Fig. 3. Risk factors causing failures of belt conveyors

As a result of preparing the Ishikawa diagram, a summary of causes of the problem (risk factors), divided into groups, was obtained, but it does not result from it, which causes contribute to the highest extent to the effect, i.e. a failure. In order to determine the extent of the impact of individual risk factors on the process of transporting excavated material by a belt conveyor, it is required to perform the FMEA analysis.

The values of the linguistic variables used to calculate the Risk Priority Number (RPN) are shown in Table 2, Table 3 and Table 4. The interval values of the variables P, Z and T represent defuzzified values of respective linguistic variables.

Table 2. Linguistic variable and its defuzzified values for the occurrence of the risk factor

Linguistic variable for the occurrence of the risk factor	Characteristics	P [rank]
Remote	A failure is improbable	1
Low	Single occurrences	2 - 3
Moderate	A failure occurs occasionally	4 - 6
High	A failure occurs with a high frequency	7 - 8
Very high	A failure is almost inevitable	9 - 10

Table 3. Linguistic variable and its defuzzified values for the effect of the occurrence of the risk factor

Effect rate	Characteristics	Z [rank]
None	No effect	1
Minor	Minor disturbances in the operation; nuisances are noticed only by some workers	2 - 3
Low	Minor disturbances in the operation; minor impact on safety; some activities are burdensome without reduction in the performance	4 - 5
Moderate	Minor disturbances in the operation, the condition affects the safety in less than 100%; working is burdensome without reduction in the performance	6
High	Minor disturbances in the operation, the condition affects the safety in less than 100%; a reduction in the performance without a loss of the equipment function	7
Very high	Significant disturbances in the operation, the condition affects the safety in 100%; a loss of the equipment function	8
Hazardous with warning	Hazardous to workers, significantly affects the safety, the condition is inconsistent with regulations and standards, the hazard occurs with a warning	9
Hazardous without warning	Hazardous to workers, significantly affects the safety, the condition is inconsistent with regulations and standards, the hazard occurs without warning	10

Table 4. Linguistic variable and its defuzzified values for the detection rate of the risk factor

Detection rate	Probability of detection of a failure by control	T [rank]
Almost certain	The process is protected against the occurrence of a failure; failures are always detected	1
Very high	Controlling and finding a failure stops the process; failures are almost always detected	2
High	High probability that the failure will be detected	3 - 4
Moderate	Control may detect the occurrence of the failure	5 - 6
Low	Control has a low chance to detect the failure	7 - 8
Very low	Control probably will not detect the failure	9
Absolute uncertainty	Control will not detect the failure	10

The FMEA analysis was prepared on the basis of the stages of the process of transportation by a belt conveyor and the risk factors presented in Fig. 3. Table 5 shows the FMEA analysis performed for two first stages of the belt conveyor operation.

Table 5. FMEA analysis of two stages of the belt conveyor operation

Operation/Process stage	Possible risk factors	Effects caused by the risk factors	Current state			
			Risk factor assessment (P) [rank]	Effect assessment Z [rank]	Hazard assessment T [rank]	RPN
Start-up of the belt conveyor	Transmission system failure	Gearbox failure	5	8	2	80
		Coupling failure	5	8	3	120
		Motor failure	6	7	3	126
		Pulley failure	6	6	5	180
	Belt damage	Belt breakage	5	9	2	90
		Belt slip-off	5	9	7	315
Loading the belt conveyor	Excavated material blocked in the holding tank	Delay in transport	7	4	1	28
	Reduction in the clearance at the trays feeding the material	Delay in transport	6	5	2	60
	Faulty operation of the feeder drive	Delay in transport, belt damage	3	5	4	60
	Lack of excavated material	Delay in transport	1	8	1	8
	The chute is set improperly	Limited discharge to the belt	4	5	4	80

Basing on the FMEA analysis, corrective actions for the transport process were proposed in order to reduce the negative impact of the risk factors. For the first stage of the operation of a belt conveyor, the proposed corrective actions are as follows: regular inspections, minor repair activities performed every day before starting up of the conveyor, such as cleaning the belt or a visual inspection of conveyor condition. Additional activities which should be performed include: determining the actual time of operation of individual drive units, analysing the vibration, current and temperature signals, and inspecting the condition of wires and their connections in the electric

motor. In addition it is recommended to take care of quality of the transported material, which means that the excavated material should be smaller in size and dry so that the clearance in the feeder is not reduced. The corrective actions performed at this stage include also taking care that the speed of feeding the excavated material onto the belt is constant. After the corrective actions has been implemented, the FMEA analysis was performed again and its part is shown in Table 6.

Table 6. The FMEA analysis after the implementation of the corrective actions

Operation/Process stage	Possible risk factors	Effects caused by the risk factors	The state after the implementation of the corrective actions			
			Risk factor assessment (P) [rank]	Effect assessment Z [rank]	Hazard assessment T [rank]	RPN
Start-up of the belt conveyor	Transmission system failure	Gearbox failure	4	8	2	64
		Coupling failure	3	7	3	63
		Motor failure	5	7	2	70
		Drum failure	5	5	4	100
	Belt damage	Belt breakage	3	8	2	48
		Belt slip-off	4	8	5	160
Loading the belt conveyor	Excavated material blocked in the holding tank	Delay in transport	6	4	1	24
	Reduction in the clearance at the trays feeding the material	Delay in transport	6	5	2	60
	Faulty operation of the feeder drive	Delay in transport, belt damage	3	5	4	60
	Lack of excavated material	Delay in transport	1	8	1	8
	The chute is set improperly	Limited discharge to the belt	4	5	3	60

After the corrective actions have been implemented, the risk of a failure in the analysed areas of the production system was reduced. The corrective actions consisted primarily in regular inspections and rigorous record-keeping, which was enough to mitigate the impact of the risk factors on the process.

4 Conclusion

Smooth operation of a production system is a phenomenon that occurs less and less often. It happens more and more frequently that the attention is drawn to the need of detecting the threats early and collecting the information concerning the cause-effect relationships occurring in the system. The FMEA analysis performed with the use of linguistic variables helped to determine the cause-effect relationships associated with the occurrence of risk factors and then minimize their impact on the production system. In the era of dynamic changes in the market environment, the FMEA method proved to be a good alternative solution that enables quick identification of potential risks for a company. When assessing a risk, linguistic variables are particularly useful, because it is possible to record the information about potential threats on the basis of verbal communications.

References

1. Burduk, A., Chlebus, E.: Methods of risk evaluation in manufacturing systems. Archives of Civil and Mechanical Engineering 9(3), 17–30 (2009)
2. Chlebus, E., Krot, K., Kuliberda, M.: Rule-based expert system dedicated for technological applications. In: Hybrid Artificial Intelligent Systems, 6th International Conference, Wroclaw, vol. 1, pp. 373–380 (2011)
3. Chrysler Cooperation, Ford Motor Company, General Motors Cooperation, Potential Failure Mode and Effects Analysis (FMEA), 1 edn. (February 1993)
4. Hamrol A.: Zarządzanie jakością z przykładami. Wydawnictwo Naukowe PWN, Warszawa (2007)
5. Mueller, D.H., Tietjen, T.: FMEA - Praxis Das Komplettpaket für Training und Anwendung. Carl Hanser Verlag, München (2003)
6. Sankar, N., Prabhu, B.: Modified approach for prioritization of failures in a system failure mode and effects analysis. International Journal of Quality & Reliability Management 18(3), 324–336 (2001)

Hybrid Methods Aiding Organisational and Technological Production Preparation Using Simulation Models of Nonlinear Production Systems

Arkadiusz Kowalski and Tomasz Marut

Wrocław University of Technology, 27 Wybrzeże Wyspiańskiego St,
50-370 Wrocław, Poland
Arkadiusz.Kowalski@pwr.wroc.pl

Abstract. Various problem solving techniques are used in organisational and technological production preparation in combinations assuring the overarching goals to be achieved in an optimum manner. The paper presents current progress in planning a facility manufacturing cabinet furniture. In order to determine output level and match a production process, expert knowledge, theoretical computations (Schmigalla method of triangles) and data aggregation were used. The entire project was then verified using adequate simulation models.

Keywords: Production preparation, modelling, simulation, hybridity.

1 Introduction

The all-important issue in planning a new production facility – in this case intended to manufacture cabinet furniture – is to determine the output level and pair it with an adequate production process. The output level is determined based on available expert and theoretical knowledge. It lays foundations for further work and computations ultimately verified using computer simulation by means of a simulation model combining discrete and continuous simulation.

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 [3], [7], [11], into account should be also taken kind of structure of the production system. 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. 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 [2], [4].

Set out were the following fundamental tasks:

- adjust the technological process to the ten-fold higher sales plan,
- select means of production,
- determine layout of workstations,

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 259–266, 2012.

- identify of the structure of production system,
- verify the project using adequate simulation models.

A system, which employs in excess of one problem solving technique, can be classified as a hybrid system. Among fundamental problem solving techniques are: data aggregation, fuzzy logic, genetic algorithms, expert systems, simulation methods, neural networks and other.

2 Characteristics of Production Processes

The enterprise had already put some effort into development aimed at expanding the range of products with cabinet furniture finished with natural wood veneer. The process was initiated by designing a collection of furniture and mocking up prototypes using available means of production.

The collection of furniture produced, are high quality cabinet furniture finished with natural beech-wood veneer, intended for dining rooms, lounges, offices and living rooms. The collection features approx. 50 pieces coming in different sizes. High variability of products without a shadow of a doubt hinders building a simulation model.

A system producing a selection of furniture, characterises with a set of features giving evidence of its non-rhythmicity (non-pipelined [1], [5]). There is no pre-determined production program, which would regulate time-wise the course of operations against a schedule. Production management requires from managers and production foremen knowledge, experience and intuition. Subsequently, both the irregularity of the production plan and application of different type random variables, proved particularly challenging to constructors of the simulation model.

Furniture is manufactured - up to the operation of dyeing - by a push system: completed pieces are stored at work in process storage. Further processing continues upon and in line with client orders. Starting from there, furniture is manufactured by a pull system.

The process can be divided into three main stages: chipboard and fibreboard processing, plywood processing and lumber processing. The process includes the following machining operations: cutting, milling, drilling, grinding and refining i.e. dyeing and varnishing, subsequently gluing and assembling.

In-process quality control takes place after each operation – machine operators are obliged to self-control. In-process transport uses industrial transport trolleys and pallets, both of which were adapted to the furniture production process.

3 Forecasted Sales Volumes

Sales volumes of new collection of cabinet furniture – i.e. production volume of finished products – were forecasted using two basic research methods:

- a quantitative method of similarity – imitation,
- a qualitative method.

The former forecasts aggregate sales volumes of products newly or lately launched to the market, based on sales figures for similar products launched earlier (qualitative method of similarity – imitation).

The later uses expert knowledge to evaluate expected sales (qualitative method), based on opinions and plans envisaged by company owners and marketing staff.

Precise future order figures remain unknown for individual pieces of furniture, thus computations were carried out for an arithmetic mean of material consumption across the entire batch of products, for a single piece. The data illustrates total material used in production, including: furniture body, drawers, solid wood doors, wooden strip doors and used auxiliary fixings: handles, guides, hinges, pegs etc. Based on the production process, opinions and experiences of the production manager, all machines and equipment required to manufacture cabinet furniture were established.

Due to ever-increasing labour costs and company's strive to assure high quality, the majority of technological operations should be automated using high-end production equipment. Such machines guarantee high: repeatability, precision, tolerances, processing speed. Moreover, they require less professional supervision, and can be operated by less qualified employees.

4 Production and Organisational Parameters

Due to technology-related imperfections, material defects, and finger trouble, defective pieces are being manufactured over the course of production, which could neither be sold nor repaired. Production plan should compensate for and accommodate rejects, so it could satisfy expected market pull. Those needs are included in the corrected production program [1], formulated as:

$$N = N_e(1 + b) \tag{1}$$

where: N – corrected production program, N_e – forecasted sales volumes, b – target level of rejects.

The b coefficient, here 0.5%, was empirically derived based on previous production runs of furniture and pieces finished with veneer. Having substituted into the equation forecasted sales, it produced 5000 pieces from the collection of cabinet furniture. Derived results play a marginal role in the increase of material consumption. Such insignificant increase is caused by low level of rejects. It stems from high-precision machining, highly qualified staff and raw materials enabling repair of possible rejects.

The factor critical to efficiency and costs of production is the minimum batch size. "A batch is a group of homogeneous pieces manufactured by a workstation at a constant set-up time, uninterrupted for manufacturing other work pieces …" [6]. The aim of estimating that parameter is to avoid having to frequently changeover equipment and to maintain flexible and multivariate production. The method of changeover share [1] is one of the methods for computing that parameter, where the minimum batch size is produced using the formula:

$$S_{ek} = \frac{t_{pz}}{q \times t_j} \tag{2}$$

where: t_{pz} – set-up time, q – empirically derived changeover loss factor, t_j – time per unit.

Taken into account were technological operations characterised by the highest set-up time to time per unit ratio. The q parameter was attributed the 0.15 value for complicated, expensive parts, which contributed a large share of finished product costs. At t_{pz} = 0,5 h and t_j = 0,05 h the minimum batch size was 67 pieces.

The available working time per employee Fr enables determining actual, planned employee utilisation for production, factoring in downtimes [1].

$$F_r = F_{nr} \times \eta_{pr} \tag{3}$$

where: F_{nr} – nominal working time per employee – h/year, η_{pr} – coefficient factoring in employee downtimes.

5 Layout Planning

Distribution of workstation within workcells is crucial to organisation and efficiency of work. A random sequence of workstations increases the length of transportation routes and causes transportation flows to cross. Those problems intensify when blue collar workers are delegated to transport the pieces.

The Schmigalla method of triangles was selected to distribute the workstations [8], [9], [10]. The salient criterion behind this method was its high accuracy coupled with computing speed. However, its drawback is inability to include real distances between workstations: distances between neighbouring equipment are fixed and equal to the grid module – figure 1.

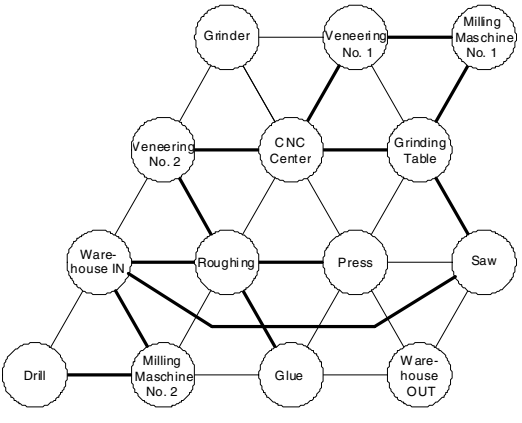


Fig. 1. General solution produced by the Schmigalla method of triangles – factory floor 1

The most important criterion is to minimise the distances between machine tools with the most frequent material flows.

Interdependencies and links between workstations and machines in the production process are illustrated by the modified depiction of the production process. Individual operations were allocated with machine tools and workstations.

6 Determining the Required Number of Machine Tools

Meeting monthly production plans would not be possible without adequate means of productions, which were determined in previous subsections. The extent, to which the plan was met, is also influenced by the number of machines, equipment and workstations. Workstation utilisation can serve as the starting parameter for determining analytically the number of required machine tools [1]. It informs about the time the machine takes to complete a production task. Global workstation utilisation T_{gk} is produced by adding preparation time and lead time, which is time-specific:

$$T_{gk} = T_{pzk} + T_{jk} \tag{4}$$

where: T_{gk} – global workstation utilisation k – of those workstations, T_{pzk} – t_{pz}-related workstation utilisation, T_{jk} – t_j-related workstation utilisation.

Bearing in mind that workstation utilisation is dictated by the production plan and batch size, that relation is illustrated with the following formula:

$$T_{gk} = \sum_k \left(n_i \times t_{pzij} + N_i \times t_{ij} \right) \tag{5}$$

where: n_i – the number of homogenous piece batches, t_{pzij} – „ij-th" operation's set-up time , N_i – production program for the i-th product t_{ji} – i-th operation's time per unit, k – type of homogenous workstations.

$$n_i = \frac{N_i}{S_i} \tag{6}$$

where: S_i – batch size. Thus the required number of workstations per cell is:

$$L^o_{mk} = \frac{T_{gk}}{F_{jk}} \tag{7}$$

where: L^o_{mk} – analytical number of workstations, F_{jk} – available working time per given type of equipment.

7 Building a Simulation Model of the Planned Production System

Simulating the facility producing cabinet furniture is intended to help achieving the following goals:

- verify the feasibility of the production plan,
- verify the analytical number of machines and workplaces,

- determine the minimum number of pallets and industrial transport trolleys,
- verify and optimise planned supply and inventory of raw materials, semi-finished products and fixings,
- target bottlenecks in the production process and machines as well as workstations of highest utilisation.

In order to achieve the above-mentioned goals, actions have to be taken to build an adequate model of the process producing cabinet furniture. Because the time from system input to output is mostly influenced by the material flows and machining times at each workstation, and on the back of an ABC analysis a decision was reached, that the entire furniture collection would be represented by a small chest of drawers and a glass panel. They were selected based on the fact, that production of each requires almost all materials and semi-finished products.

Building a simulation model entails defining workstations, produced pieces, transportation routes and manufacturing resources. Then, modelled are production processes, deliveries, stoppage and shifts in the production system. Then, defined are variables, macros, arrays, sub-processes, distributions, attributes etc. Their combination should help to best represent the complex reality. A model combining features of discrete and continuous simulation achieved the desired result. The simulation model built in that manner was subject to simulation analysis. Subsequently, it was verified and validated as well.

8 The Experiment

Dry runs of the simulation experiment were being carried out since early stages of the model building, to find errors and verify it against reality on a regular basis. Figure 2 illustrates a part of described simulation model.

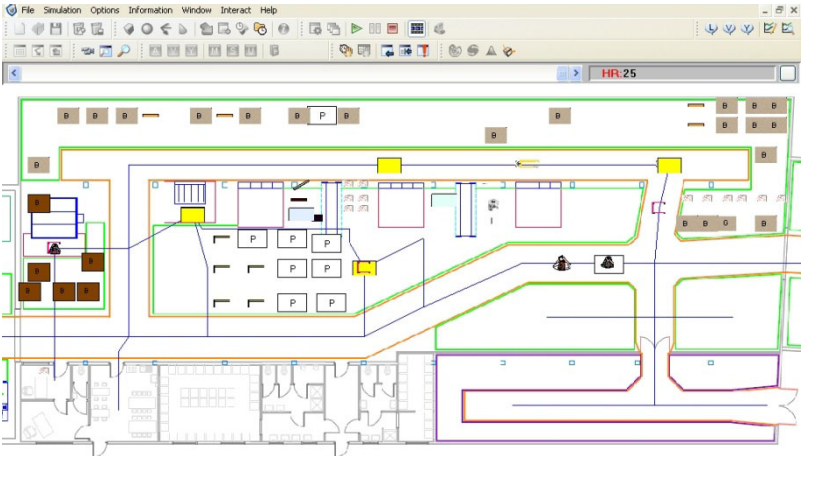


Fig. 2. A part of the model simulating the process producing cabinet furniture

The optimum number of workers was determined over two stages. At the first stage, workers were organised into groups by three factory floors and varnishers. Each worker had an assigned workstation, and could complete one's tasks within the designated work zone. Obtained data on utilisation of workstations and particular groups of workers allowed targeting utilisation hotspots. Bottlenecks – i.a. format effector, CNC machining centre, veneering machines for narrow pieces, painting line, floodbar – were all assigned with individual workers. That modification brought higher productivity and shortened the time required to produce planned selection of furniture.

At that stage the simulation was run iteratively. Additionally, after each simulation run results were analysed. Hence a desirable solution could be found, which was in line with experiment goals. The simulation time was defined as 2 working months (353 h) in order to obtained more repeatable results.

9 Analysis of the Results

Having configured the simulation model as discussed, the production plan was met in 99%. The first experiment goal i.e. "verify the feasibility of the production plan", was considered achieved.

The second goal i.e. "verify the analytical number of machines and workplaces", was achieved as well. The number of machines and workstations guarantees the production program to be met. After the results were analysed, there was no need to modify neither the number of workstations nor machines.

Based on curried out simulation, studied results and the experience in producing furniture, the number of blue collar workers came under scrutiny.

The proposed level of employment guarantees the facility to hit its target efficiency, and to keep employment-related costs low. The minimum number of industrial transport trolleys and pallets was approached similarly. Excessively low number of means of transport would jeopardise efficiency, by causing queues at workstations and by blocking machines, whereas their excess would generate additional costs and create the need for storing areas for redundant units. Based on the simulation model it was deduced, that assembly workstations show the highest utilisation percentage, caused by pieces awaiting other components. The time computed in that manner did not match the analytical working time.

Drawing on results, it can be concluded that there are production capacity reserves at factory floor number 1, which are currently constrained by the production program aligned with planned facility efficiency. Production efficiency at factory floor number 2 is constrained by the flow of semi-finished products from factory floor number 3 – furniture hold the assembly workstation long, waiting for components. Bottleneck at the factory floor number 3 is the printing line, which essentially constrains efficiency of the entire facility.

Further improvements of the simulation model could entail introduction of prioritised batches most needed at a specific point in time. Such solution would bring the model ever closer to an actual production system controlled by a production manager.

References

1. Brzeziński, M. (eds.): Production organisation and management, planning of pro-duction systems and production management processes, 152, 41, 156, 51, 48, 161. Placet Publishing House, Warsaw (2002)
2. Burduk, A.: Evaluation of the Risk in Production Systems with a Parallel Reliability Structure Taking into account its Acceptance Level. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS, vol. 6679, pp. 389–396. Springer, Heidelberg (2011)
3. Chlebus, E., Burduk, A., Kowalski, A.: Concept of a Data Exchange Agent System for Automatic Construction of Simulation Models of Manufacturing Processes. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS (LNAI), vol. 6679, pp. 381–388. Springer, Heidelberg (2011)
4. Chlebus,. E.: CAx Computer Techniques in production engineering. WNT, Warsaw (2000)
5. Papadopoulos, C.T., Vidalis, M.J., O'Kelly, M., Spinellis, D.: Manufacturing Systems: Types and Modeling, Analysis and Design of Discrete Part Production Lines. Springer Optimization and Its Applications. LLC (2009)
6. Pasternak, K.: Production management outlined, 178, 144. Polish Economic Publisher, Warsaw (2005)
7. Ryan, J., Heavey, C.: Process modeling for simulation. Computers in Industry 57, 437–450 (2006)
8. Schmigalla, H.: Factory planning: concepts and connections. Hanser, München (1995)
9. Schmigalla, H.: Methods for optimal machine configuration. Verlag Technik Berlin, Berlin (1970)
10. Schmigalla, H.: Computer-aided Designing: Operational design of dialogue. Verlag Technik Berlin, Berlin (1986)
11. Wenzel, S., Boyaci, P., Jessen, U.: Simulation in Production and Logistics: Trends, Solutions and Applications. In: Dangelmaier, W., Blecken, A., Delius, R., Klöpfer, S. (eds.) IHNS 2010. LNBIP, vol. 46, pp. 73–84. Springer, Heidelberg (2010)

The Concept of Intelligent System for Horizontal Transport in a Copper Ore Mine

Tomasz Chlebus[1] and Pawel Stefaniak[2]

[1] Wrocław University of Technology
[2] KGHM Cuprum Ltd Research and Development Centre
tomasz.chlebus@pwr.wroc.pl, p.stefaniak@cuprum.wroc.pl

Abstract. The Article presents the concept of intelligent transportation system, which could be implemented in copper mines. The task of such a system is intelligent and safe management of the horizontal transport by means of band conveyors. The solutions described here are aimed at showing how the transport could be automated, the problems signalled and the modern band conveyor systems utilised optimally.

Keywords: Mining, Conveyor, transportation, ore.

1 Characteristics of the Transport Infrastructure in Copper Ore Mines

Horizontal transport in copper ore mines consists mainly in displacing the winning by means of band conveyors. In most of the Lower Silesia mines the band transportation of winning has been supported by railway systems and mining machines taking part in the initial phase of the ore mining.

The chamber-pillar winning system being operated in the mines forces application of the specific system of transportation. Specificity of the horizontal transport in mines consists in using at various stages and on the various scale the cyclic transport means (automotive machines, underground mine railway), as well as the continuous transport of high capacity (band conveyors) [1]. At the first stage of winning the ore the bucket loaders and haulage vehicles are used. In case the distance from the winning place to the first reloading point – the chute grate, witch dimensions are about 300x3000 cm, (Fig.1) [2] is small (up to 250m), the won ore is carried by large bucket loaders.

In case of larger distances, or in situations when a small loader is operating at the mine face, the haulage vehicles filled by the loader are used for transportation. The ore carried by the cyclic transport means is poured on to a grate, where it falls to a chute making the reciprocating movement, and further to a band conveyor. The chute grate has a mesh of 450 mm, which allows for passing of ore pieces with sizes safe for the band conveyor (Fig.1). In case the lumps of winning are too large, they are subjected to size reduction in a crusher. Only one Lower Silesian mine is utilising the bend conveyors exclusively (Fig.3).

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 267–273, 2012.

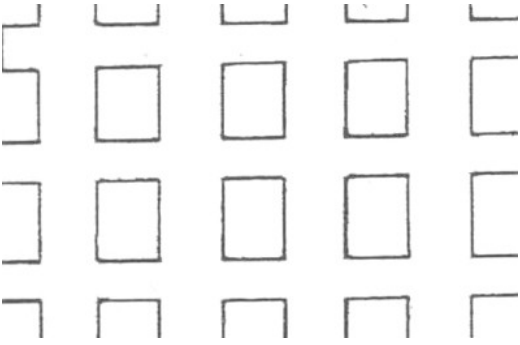


Fig. 1. The chute grate in a copper ore mine

Remaining mines, more or less, support their activities with the underground railway systems. One of the most serious problems in the copper ore transportation is discontinuity in supplying the winning to the grate and a resultant cycling in the band conveyors loading. Such non-uniformity is related to the local overloading of bands and a consequent pouring out of the winning or damaging the conveyor.

In one of the mines the transport is mainly accomplished by means of the underground railway (Fig.2). A sequence of transportation phases is as follows: winning of ore – transport to a grate – band conveyor – railway transport – dump.

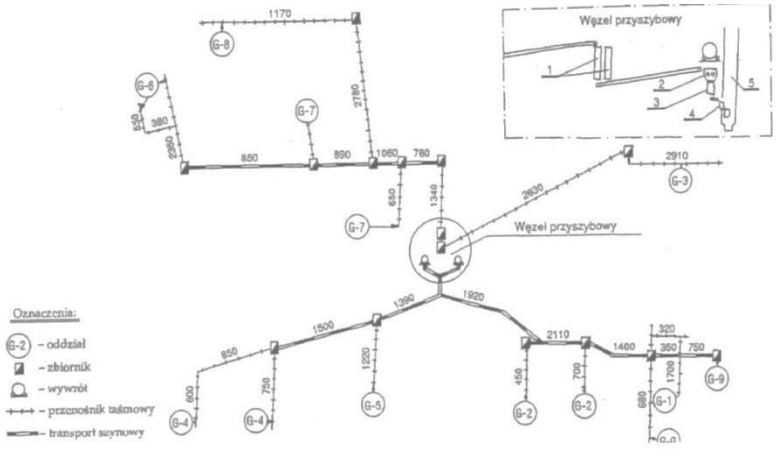


Fig. 2. Lubin mine [2, s.575], Designations: G-2 - Section number; [/] - Winning Collector; O – Dump;+++ - Band Conveyor = = = - Railway transport

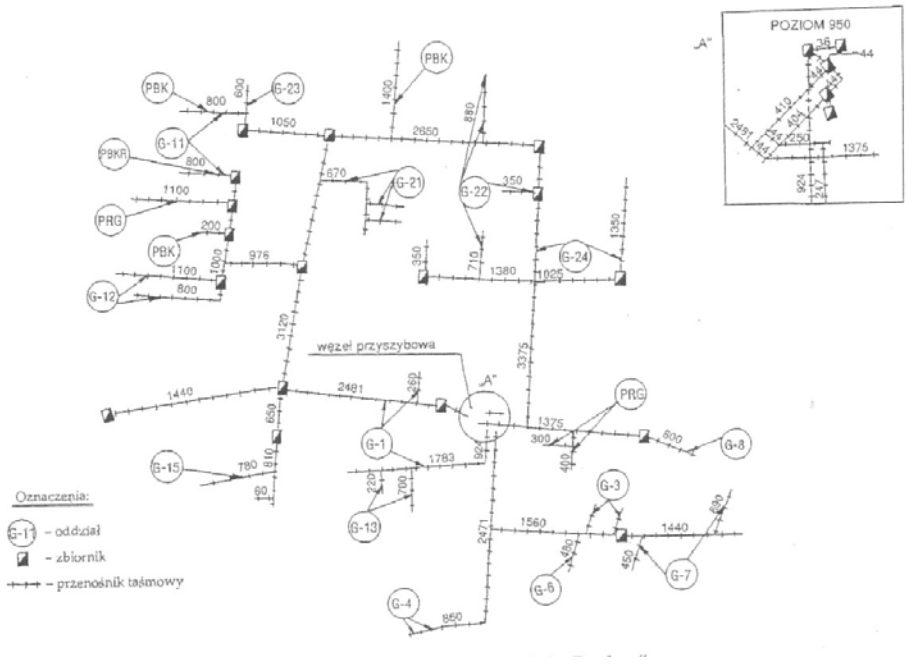


Fig. 3. Exemplary topography of band transporter in a mine, Designations: G-11- Section number; [/] - Winning Collector; +++ Band Conveyor

2 A concept of Implementing the Intelligent Suspension System for Roller Sets in Band Conveyors

Traditionally, the suspension of rollers in band conveyors is rigid and it does not react to the varying load appearing at the times the winning hauling vehicles approach the chute point. It is true that the task of chute is fairly uniform distribution of ore on a band conveyor however there may be cases when a significant quantity of smaller pieces of ore could be delivered to the grate, resulting in sudden load of a short section of the band.

Significant loads influence faster wear of the band and rollers, raising that way the winning costs resulting from the need of frequent exchange of the operating parts. In a typical mine within the 24 hours period a load of 23600 Mg (mega grams) of ore and about 700 Mg of materials is transported. Such loads set high strength requirements for the transportation system, which is supposed to operate the whole day round.

Research of the real conveyors capacity [3] has shown 80% utilisation of their full capacity. Besides that, in cases of particularly long sections of the conveyors the rollers support a band without winning. In such cases no force is acting on the rollers, which causes significantly lower tension of the band than in case of the loaded conveyor. Besides the band tension, also the angle of cavity, which could be used with various streams of winning, could change. With low load the cavity angle can be

small, and instead, along with the growth in load the cavity angle should increase causing that way an increase in the transport safety and a decrease in rollers loading.

Such solution could be achieved by suspension of rollers, which could change the geometry according to its load (Fig. 4). The solution could be achieved by using the elastic lifting slings.

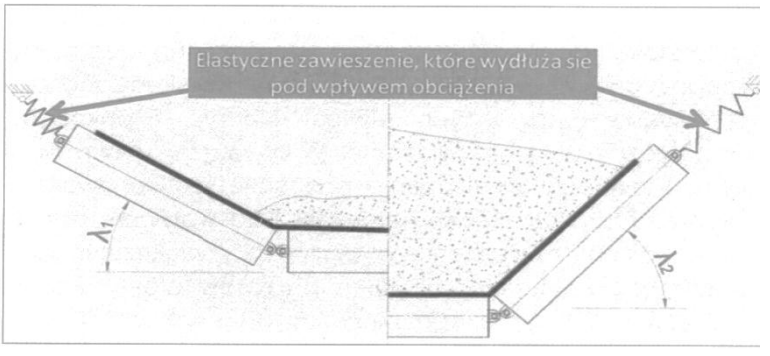


Fig. 4. Elastic suspension elongating with the load increase [4]

In such a solution the suspension remains short creating the low cavity angle while the load is low, and instead, under the increasing load with winning the suspension lengthens increasing the cavity depth. The concept is not the new one. The first patent for the elastic suspension comes from 1909. Each solution refers to the system with repeatable parts. The Artur Küpper GmbH Company has developed the new solution, which could be applied to the existing bearing elements of the route. Moreover, the concept assumes the use of standard hooks, so as not to complicate the assembling and disassembling processes of the roller sets (Fig.5). The concept of Artur Küpper GmbH Company consists in fixing the rubber springs on ropes in places where the rigid suspension appeared so far [4].

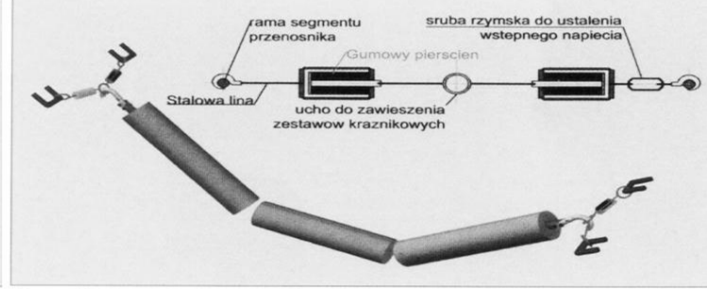


Fig. 5. Technical solution of the elastic suspension proposed by Artur Kupper GmbH[4]

By proper selection of the rubber spring parameters related to the suspension geometry the optimum relationship between the elongation and the load could be

achieved. More precise elaboration of the roller and spring system able to reflect the specific mine and copper ore load conditions should be calculated using such computational systems as QNK-TT, where the load non-uniformities could be forecast and the sets selected so, as not to deflect too much on either side. Such solution would enable significant elongation of the roller sets life.

3 Telematics Transport System in Ore Mine

Subsequent element rationalizing the work in copper ore mine should be automation and development of the intelligent system for horizontal band transportation control. Such solution should be based on the set of sensors and conveyors control system. So far, the approval for entry on the grate is issued by a miner working at the URB – a unit crashing the winning (Fig.6) taking into account business degree of the grate.

With the proposed solution a decision on entering the grate field by a carrier vehicle or a loader would also be taken by an URB operator, but beside the information on the degree of filling the grate the electronic information system which continuously checks the status of the transport system would supply him additional information by means of sensors such as a sensor of conveyor band convergence, a sensor of a conveyor band movement, on the degree of filling the collecting vessels localized at the far end of the conveyors, as well as the load and operation of chute, at which in th most cases the winning material is always present.

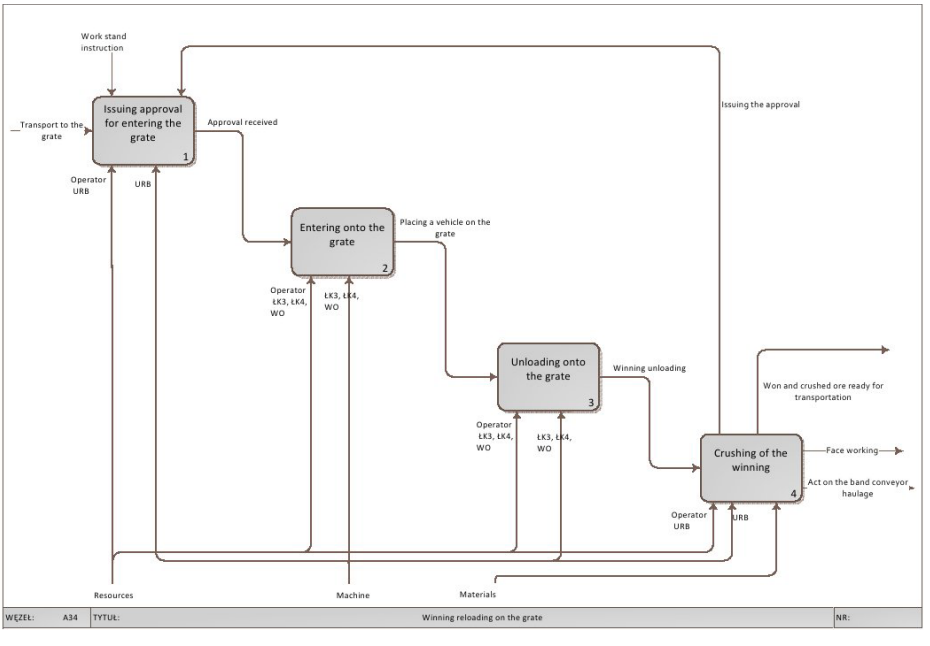


Fig. 6. Classic approach to the chute management

The very important aspect that has to be considered by the intelligent transport system is a sensor, or a set of sensors, checking the content and concentration of ore weight being displaced by the band conveyor. Such a sensor should be located possibly closest to the chute in order to detect the elements protruding out of the band, and which could create a danger for people and machines located in the nearest proximity of the band conveyor and, depending on the measurement result from the mentioned sensor, to stop or continue the conveyor operation. In the proposed solution an algorithm appears, which should be implemented in the intelligent system of transportation control (Fig.7). The information obtained from the system is short in form, and the operations to be performed at the stage „System checks the band conveyor status" should check the full range of sensors and elements of the system so, as to relieve the operators from making the decision concerning continuation of work, which in the significant degree is not observable for him.

Exact algorithms are not presented here because of the diversified specificity of each mine. In order to simplify implementation and management of the intelligent transport system, the logic of conduct should be the same for each mine. Of course, the information obtained from multiple sensors should be currently analysed and, similarly as the information supplied from the sensors of the storage containers or the chute loading sensors for example, could be determined with the use of discrete methods, and the information on sliding/dropping from band, from track or on the overload should be delivered continuously in order to provide for optimal track of the material stream and wear of the transportation system rollers. Additional part of the system is supervision of the transport ways so, as to continuously visually control

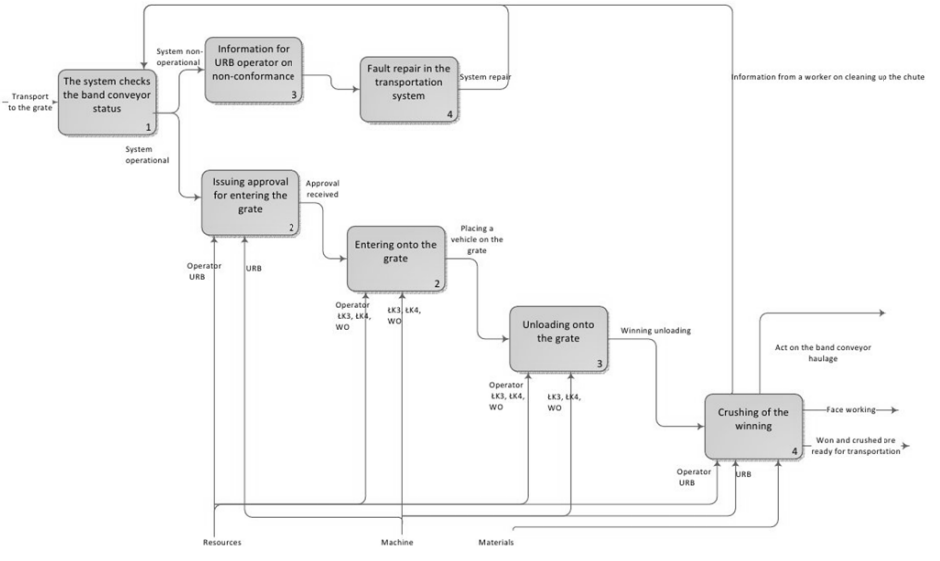


Fig. 7. Intelligent solution for a transportation system control

the operation. Application of industrial cameras for that purpose would enable detecting improper behaviour of workers and of the intelligent transportation system. Such supervision should be continuous in nature because of the possible earth movements and breaks in the system operation, not necessarily appearing in the right time noticed by the system.

4 Summary

Optimization of the contemporary horizontal transport systems in copper ore mines should follow towards eliminating the chance for human error. So far as the operations preceding transportation such as drilling, winning, shooting are sanctioned by the mining law regulations, so much the systems of horizontal transportation, especially the band one, are designed so as to best utilize the possibilities and limitations of the chamber-pillar mining system. Modification of the rollers suspension in band conveyors should significantly lengthen the cycle between the operating parts exchange.

References

1. Industrial Transportation and Working Machines No (3) (2009) (in Polish)
2. Monograph of KGHM PolskaMiedź S.A. (in Polish)
3. Geesmann, F.O.: Experimentale und TheorethischeUntersuchungen der Bewegungswiederstaende von Gurtfoardenanlagen. Diss. Uni. Hannover (2001)
4. Industrial Transportation and Working Machines No (1) (2008) (in Polish)

Integration Production Planning and Scheduling Systems for Determination of Transitional Phases in Repetitive Production

Damian Krenczyk, Krzysztof Kalinowski, and Cezary Grabowik

Silesian University of Technology, Institute of Engineering Processes
Automation and Integrated Manufacturing Systems,
Konarskiego 18A, 44-100 Gliwice, Poland
{damian.krenczyk,krzysztof.kalinowski,cezary.grabowik)@polsl.pl

Abstract. In the process of multiassortment repetitive production planning the designation of operating system parameters for the steady state (in which after the last operation the return to the first operation in the sequence of productive resources occurs) is required. The length of steady state of the system is determined by the work of the critical resource. Determination of resources sequence for the steady state gives possibility to determine the processing times for processes in the system without taking into account the transitional phases. During the execution of this type of production, in addition to the phase in which production takes place in the steady state, transition phases connected with starting (start-up phase) and finishing work in the system (cease phase), and the phases associated with the changes in ongoing sequences in steady states can be distinguished. The article presents the problem of estimating the duration of the start-up phase and the cease phase in the concurrent multiassortment production systems, in which access to the resources is regulated by the local dispatching rules. A method and a procedure of creating a schedule for the transition phases using the integration between the systems of production planning and scheduling is presented. The demonstrated approach is illustrated with the examples using KbRS and SWZ systems.

Keywords: Production planning, integration, scheduling, xml, constraint satisfaction.

1 Introduction

Modern manufacturing systems undergo constant changes associated with the rapid advances in the field of automation, the introduction of new means of production, development of new technologies and the use of new ways of production planning and control, aided by rapidly developing information systems. Computer aided techniques, due to continuous reduction of their prices, are becoming easily available and more commonly used in the preparation of organizational stages of the production in SMEs [4, 9]. The necessity to use computer aided techniques results from the fact that in these production systems production is often executed as a concurrent repetitive

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 274–283, 2012.

multiassortment production, characterized by producing diverse range of products at the same time and also by flexibility. This causes difficulties in meeting the control parameters, which, for this type of production, include: date of execution of production orders, use of resources and ensuring an acceptable quality of the system (without deadlocks and starvations). One of the areas associated with the use of information technology is computer aided decision making, concerning the possibility of order execution on the basis of known specifications and characteristics of the production order and available production system.

Multiassortment repetitive production is characterized by the simultaneous realization of various products. In this type of system, regular and steady repetition of the operations of the manufacturing process performed on the system resources, in which after the last operation the return to the first operation in the sequence called a fixed period of the course, occurs. The length of the period is determined by critical resources [1, 2]. During the execution of this type of production, in addition to the phase in which production takes place in the steady state, transition phases can be distinguished. Transitional phases allow the synchronization of the flow of production involving the coordination of the interaction of two or more processes in order to bring them to the desired steady state and the transition between two known but different steady states (Fig. 1) [2, 3].

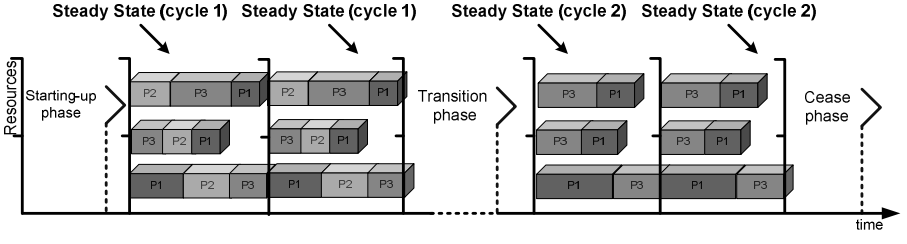


Fig. 1. Multiassortment repetitive production

Multiassortment repetitive production planning and control problems for steady and transient states have been considered in works[1 - 3]. A rapid decision making method on the acceptability of a production order, which allows quick verification of submitted orders and automatic generation of distributed control procedures, has been proposed. It has been assumed that the production flow control will be carried out under local dispatching rules, performed periodically. Local dispatching rules determine the sequence of processes, the access to the resource and their number, and provide at least a single execution of operation belonging to each process undertaken on shared resources of the system during one cycle of work rules. The access to resources is regulated according to the mutual exclusion mode. This means that the steady states are generated by sets of local dispatching rules assigned to resources. Since the work of machines and equipment is carried out according to rules generated by the cycle, its characteristic feature is that important normatives of production can be defined in an algebraic way. This gives the possibility to form a set of indicators of

production at a high level, such as resource utilization and inventory levels of work in progress [2, 3].

Since the considered steady states are generated by fixed sets of local dispatching rules (LDRSS) assigned to the systems shared resources, the natural way was to propose a set of rules for determining the transition phases, which will enable the transition from the selected initial state to one of the states of the steady state. Achieving the steady state ensuring on-time completion of production, requires the adoption of the correct sequence of processes in the rule or the execution of a specific sequence of processes (other than in the steady state), which is called the start-up rule (SR). Similarly, for cease phase (CR) (end of production) and change of the state associated with the change of the processes performed concurrently (finishing some of them, and beginning the others) [2, 3].

Execution of SR causes the initial fill of interresources buffers, in order to ensure global viability of the production system and its synchronization. By analogy to the starting-up phase, set of cease rules (CR) are determined, to allow the removal of the elements introduced in the starting-up phase from the system. Combination of local dispatching rules with SR and CR in one meta-rule (1) forms a complete set of procedures for distributed control system of concurrent processes [2]:

$$R_i=\{(SR), (LDRSS),(CR)\} \tag{1}$$

Construction of the meta-rule begins from designation of a local dispatching rule, which determines the rest of its components, ie, SR, and CR.

The method of determining meta-rules is based on the determination of meta groups of artificial intelligence methods that guarantees obtaining an acceptable solution, contrary to the methods of seeking the optimal solution, such as constraints satisfaction and DFS (depth-first search (DFS) algorithm with backtracking) described in [4].

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 [4, 5]. The fulfilment of all conditions (their conjunction) guarantees the possibility of order execution (Fig. 2). 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.

Sufficient conditions, which are used in described methodology, have been designated for the production system and production order identified constraints. The conditions include: [3, 4]:

- 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.

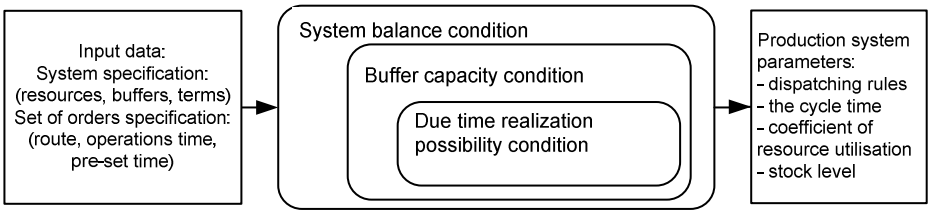


Fig. 2. Procedure of the acceptance of the production orders set for realisation in the system [4]

Discussed support rapid decision-making methodology has been implemented in orders verification computer system SWZ [2, 4].

2 Problem Formulation

The order realization time in the repetitive production system taking into account the start-up and cease phase for the Pj process can be determined from the following:

$$tk_j = tr_j + \frac{I_j - F_j^R}{F_j} \cdot T_c + T_R + T_W, \tag{2}$$

where:

tk_j - real achievable due date for completion of the P_j process,
tr_j – start-up phase time of the P_j process,
I_j - the lot size of the order,
F_j - batch size of the j-th order (the number executed during one cycle),
F_j^R - number of pieces of the j-th order executed during start-up and cease phase,
T_R – start-up rule duration time,
T_W - cease rule duration time,
T_c – cycle execution time in the steady state.

Duration of the rules execution for steady states is determined by working time of the critical resource and its value is equal to the cycle time of the system. Therefore it's able to be determined as the sum of execution times of operations on a critical resource in accordance with the designated order of operations execution by local dispatching rule.

$$T_C = \max\left(\sum_{w=1}^{o_i} \left(t_{p_{iw},i} + tpz_{p_{iw},i} \right) \right), \tag{3}$$

where:

T_c – cycle execution time in the steady state,
$t_{p_{iw},i}$ - the cycle time of the Pp_{iw}-process on the i-th resource,

$tpz_{p_{iw},i}$ - set-up time of the Pp_{iw} process on the i-th resource,

o_i- number of operations in local dispatching rule allocated to i-th resource.

Determination of duration time of transient phases (starting-up and cease) becomes a problem, because for these phases it is necessary to take into account the times associated with the consequence of the operations - waiting for products that must first be processed on different system resources. In this case, to determine the exact duration of start-up phases, production schedules for resources should be designated. At the stage of determining the rules controlling the operation of resources, it is only possible to estimate the maximum duration of the start-up rule (cease) T_R (T_W) , which (taking into account the most adverse scenario of the consequences of the operation) is not greater than the sum of operation times of all processes occurring in the rules for start-up (cease) assigned to the system resources for all resources:

$$T_R = \sum_{i=1}^{m} \sum_{w=1}^{o_i} \left(t_{p_{iw},i} \cdot K_{iw}^{R} + tpz_{p_{iw},i} \right) \tag{4}$$

$$T_W = \sum_{i=1}^{m} \sum_{w=1}^{o_i} \left(t_{p_{iw},i} \cdot K_{iw}^{W} + tpz_{p_{iw},i} \right) \tag{5}$$

where:

$t_{p_{iw},i}$ - the cycle time of the Pp_{iw}-process on the i-th resource,

$tpz_{p_{iw},i}$ - set-up time of the Pp_{iw} process on the i-th resource,

p_{iw} - numbers of processes assigned to the i-th resource, in accordance with local dispatching rule,

K_{iw}^{R} - times of occurrence of each of the processes in the starting-up rule of i-th resource,

K_{iw}^{W} - times of occurrence of each of the processes in the cease rule of i-th resource,

m - number of resources,

o_i- number of operations in local dispatching rule allocated to i-th resource.

It is the most adverse scenario. In practice, for many cases T_R and T_W will be shorter than calculated this way, because the consequences of operations for the processes occurring in start-up rule have not been taken into account.

The solution to the considered problem is to integrate the SWZ system with the production scheduling and rescheduling system KbRS, which for designated start-up and cease rules would create schedule for transitional phases, and on its basis T_R and T_W will be determined.

3 Construction of Schedule for the Transition Phase

Procedure for determining the schedule for start-up and cease phases [1, 2] has been implemented in the KbRS (Knowledge-Based Rescheduling System) system (Fig. 3),

which supports processes scheduling and rescheduling in discrete manufacturing systems [6, 7, 8]. The main task of the KbRS system is production scheduling (planning activities in time), which will ensure satisfaction of the demand for manufactured goods (execution of orders) with the best possible utilization of these resources with taking into account accepted evaluation criteria. Schedules are created by processes scheduling algorithms according to established rule of priority (LPT, SPT, EDD, etc.) and given order of operations on resources (local dispatching rule). After entering or updating the required input data and parameters defining the scheduling process, variants of the initial production schedule are generated. Production schedules are presented on Gantt charts and evaluation indicators summarized in the tables provide an overview and analysis of individual schedules.

A multi- assessment module supports selection of a schedule for execution. A list of different variants of the schedule, which in the process of searching for solutions received the highest scores is proposed by the system. The final decision on the selection and application of a schedule is taken by a planner, who in a given situation can apply additional evaluation criteria, formalized in the system. Adjustments to the generated schedule can also be made. Rescheduling process is started after registering the event (noise) or upon request, within a specified period. Schedules are calculated for the maximum, average and aggregate values of basic parameters such as: execution time, flow time, lateness, and delays.

The use of the KbRS system to determine schedules for transient phases required integration of KbRS and SWZ systems. The integration process through the use of data exchange modules that use the Extensible Markup Language XML [10, 11] (Extensible Markup Language) and developed XSLT [12] (Extensible Stylesheet Language Transformations) documents has been performed. Spreadsheets are used for XSLT processing of XML files containing the output of the SWZ and KbRS systems into input files. Integration is carried out based on the user interface developed using XSLT processor features (Fig. 3).

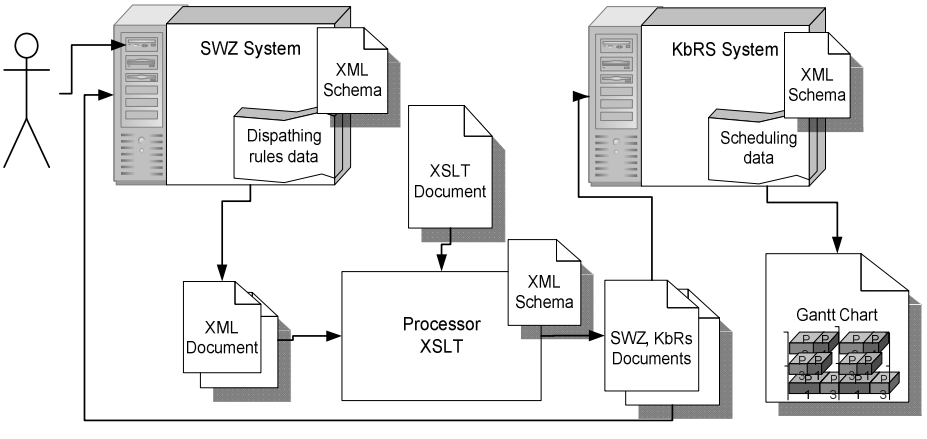


Fig. 3. SWZ and KbRS systems integration

Operation of KbRS system generating schedules for transient states is presented in Fig 4. The result of the KbRS system operation is real work schedule of the system in the transitional phases together with the designated labour indicators, ie: the actual duration of the start-up and cease phase and the coefficient of the system resources utilization.

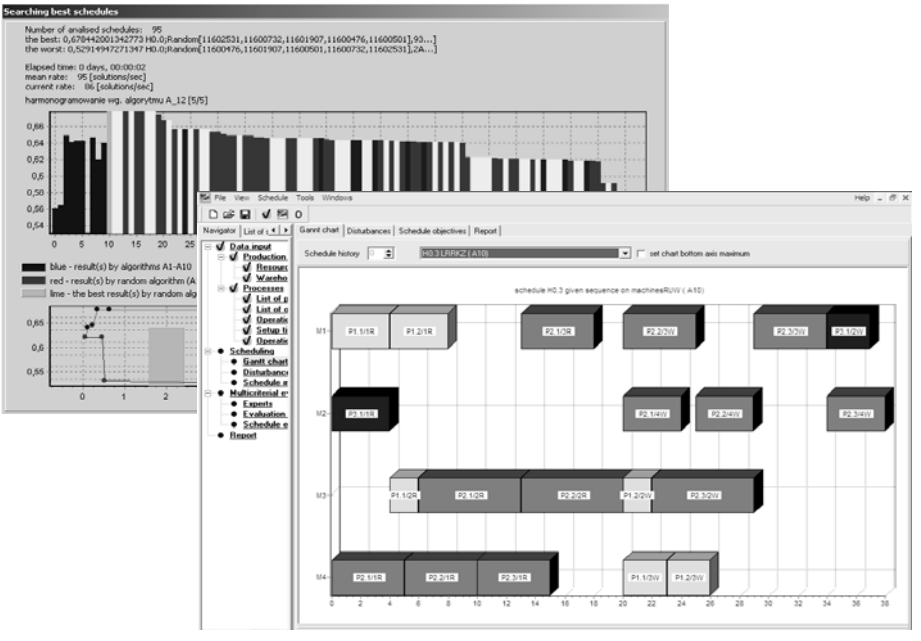


Fig. 4. KbRS system scheduling module

4 Practical Example

A production system given consists of ten resources M_1 – M10. For the execution in the system processes P_1, P_2, P_3 and P_4 are waiting. Routes are shown in Figure 5, the times are recorded in the processes matrix:

$$P_1 = \begin{bmatrix} 2 & 7 & 6 & 1 & 4 & 3 \\ 12 & 11 & 8 & 4 & 11 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; P_2 = \begin{bmatrix} 9 & 1 & 5 & 2 & 10 \\ 8 & 3 & 4 & 11 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; P_3 = \begin{bmatrix} 8 & 4 \\ 12 & 7 \\ 0 & 0 \end{bmatrix}, \; P_4 = \begin{bmatrix} 3 & 8 \\ 6 & 3 \\ 0 & 0 \end{bmatrix}.$$

The values of the first row of the matrix correspond to the resources over which the route of the process goes. In the second line cycle times on these resources are given. The third line contains the set-up times.

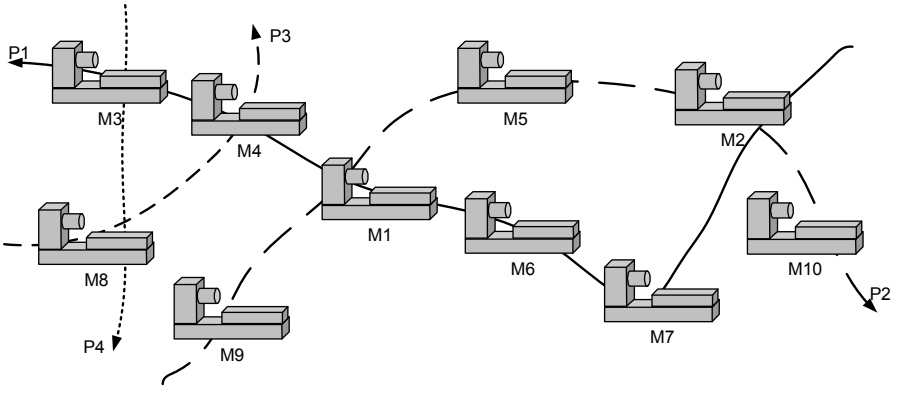


Fig. 5. Production flow

Data on production resources, and processes waiting for execution in the system were introduced to the SWZ system and meta-rules controlling operation of the system in starting-up and cease phases and in a steady state have been generated:

```
R1={(2,2,2,1,1);(1,2);(2,1,1,1)},
R2={(1,1,1,1,1,2);(1,2);(2,2,2)},
R3={(4);(1,4);(1,1,1,1,1)},
R4={(1);(1,3);(1,1,1,1,3)},
R5={(2,2);(2);(2,2)},
R6={(1,1,1);(1);(1,1)},
R7={(1,1,1,1);(1);(1)},
R8={(3);(3,4);(4)},
R9={(2,2,2,2);(2);()},
R10={();(2);(2,2,2,2)}.
```

Then the data about the production system and the rules were exported to KbRS program and work schedules for the system start-up and cease phases were generated. Duration of start-up and cease phases calculated in the SWZ (most unfavourable scenario), based on equation (2) and (3) is, respectively: $T_R = 225$, $T_W = 195$. For the designated schedule in the KbRS program these times, read from the graph, are respectively $T_R = 71$, $T_W = 51$ (Fig. 6).

As it has been expected the real duration of transition phases is much smaller than for the most adverse case, however, due to the specific execution of the transition phases it is not possible to determine the exact time of transition phases without developing a schedule for these phases.

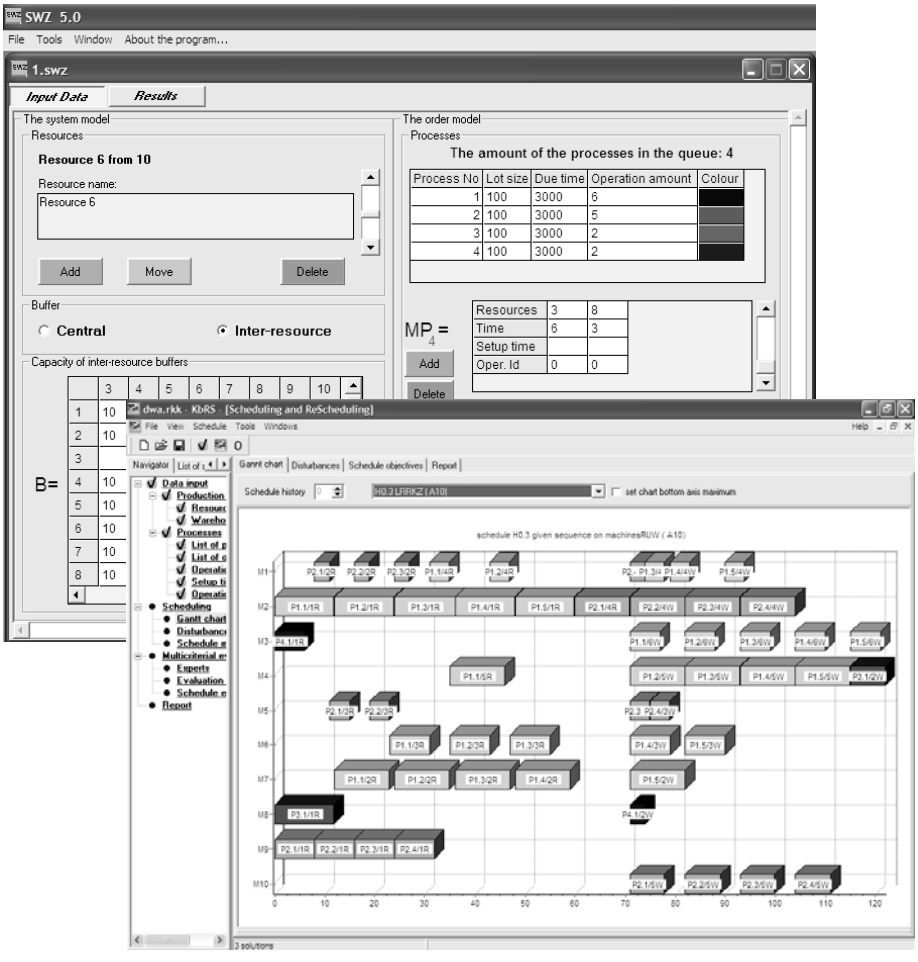


Fig. 6. SWZ calculation results and KbRS schedule

5 Summary

Integration of SWZ planning systems and a KbRS system of scheduling, carried out on the base of XML and XSLT language allows to determine the exact duration of start-up and cease phases and transition phases in repetitive multiassortment production systems. This allows to answer the question related to the lead times of production orders (including both phases of steady state and transition phases) with greater accuracy than in case of estimating the duration of the transitional phase, assuming the least favourable scenario, connected with the consequences of operations execution. In addition, the proposed method allows the visualization of these phases in the form of Gantt charts. In further studies, extending the proposed method to the module associated with the determination of parameters of repetitive production for the assembly systems is expected.

References

1. Skolud, B.: Deadlock avoidance in systems of multiassortment repetitive production. In: IFAC Workshop Series Intelligent Manufacturing Systems, pp. 245–250 (2003)
2. 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)
3. Krenczyk, D., Dobrzanska-Danikiewicz, A.: The deadlock protection method used in the production systems. Journal of Materials Processing Technology 164, 1388–1394 (2005)
4. Krenczyk, D., Skołud, B.: Production preparation and order verification systems integration using method based on data transformation and data mapping. In: Achterberg, T., Beck, J.C. (eds.) CPAIOR 2011 Part II. LNCS, vol. 6697, pp. 297–404. Springer, Heidelberg (2011)
5. Bartak, R., Salido, M.A.: Constraint satisfaction for planning and scheduling problems. Constraints 16(3), 223–227 (2011)
6. Kalinowski, K., Skołud, B., Grabowik, C., Krenczyk, D.: Computer aided technological and organizational processes planning. In: Proceedings of the Contributions of 15th International Scientific Conference, CO-MAT-TECH 2007, Quality Assurance Of Products, Safety Of Production And Environment, Trnava, Slovakia, pp. 173–176 (2007)
7. Kalinowski, K.: Scheduling of production orders with assembly operations and alternatives. In: Proc. Int. Conf. Flexible Automation and Intelligent Manufacturing, FAIM 2009, p. 85. University of Teesside, Middlesbrough (2009)
8. Kalinowski, K.: Decision making stages in production scheduling of complex products. Journal of Machine Engineering 11(1-2), 68–77 (2011)
9. Saniuk, S., Saniuk, A.: Production orders planning in a network of small and medium-sized enterprises. In: Lewandowski, J., Jałmużna, I. (eds.) Contemporary Problems in Managing Production and Services Supporting Manufacturing Processes, pp. 31–38. Wydawnictwo Politechniki Łódzkiej (2009)
10. Sormaz, D.N., Arumugam, J., Harihara, R.S., Patel, C., Neerukonda, N.: Integration of product design, process planning, scheduling, and FMS control using XML data representation. Robotics and Computer-Integrated Manufacturing 26(1), 583–595 (2010)
11. Extensible Markup Language (XML) 1.0 5th edn. W3C Recommendation (2008), http://www.w3.org/TR/2008/REC-xml-20081126/
12. XSL Transformations (XSLT) Version 2.0, W3C Recommendation (2007), http://www.w3.org/TR/xslt20/

The Hybrid Method of Knowledge Representation in a CAPP Knowledge Based System

Cezary Grabowik, Damian Krenczyk, 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,Damian.Krenczyk,
Krzysztof.Kalinowski}@polsl.pl

Abstract. In this paper the hybrid method of manufacturing knowledge representation in the CAPP knowledge based system is presented. In the presented solution the manufacturing knowledge as well as the product design is represented by means of the object method. This paper shows only results obtained at the object design stage (OOD). Currently the scope of decision problems solved by proposed system is limited to axi–symmetric products, but basis on the achievements to date it is thought to develop this system in order to widen the subjected products group. Moreover the detailed description of the inference engine of the CAPP system is given. In the inference engine the two groups of rules were distinguished. In the paper content the examples for each group of rules are shown.

Keywords: computer aided process planning (CAPP), computer aided planning (CAP), scheduling, rescheduling, knowledge representation, inference engine.

1 Introduction

The increase in competitiveness extorts in an industry application of the new methods and techniques both in the production preparation process, and production process realisation course. It is the main cause of permanent development. The manufacturing process planning is one of the most important action in the domain of the technical production preparation [1, 2, 3, 6, 9]. The main goal to achieve during this stage of the technical production preparation is to define the complete structure of the manufacturing process; it means the sequence of the manufacturing operations which have to be applied in order to transform the semi-finished product into the final product which performs all necessary design conditions in relation to surface quality, dimensional accuracy, shape accuracy etc. The manufacturing process plan should also ensure the minimal manufacturing cost, minimal labour consumption, but taking into consideration the available in the factory manufacturing recourses [1, 2, 4]. In practise the manufacturing process structure, it is the number of the manufacturing operations from the one hand depends on the machine park installed in the factory on the other hand on knowledge, skills, and professional habits of the process engineers. In the traditional approach the manufacturing process plan is made manually by the

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 284–295, 2012.

process engineer, so it can be observed strict dependency between the manufacturing process plan structure complexity and the process engineer. Taking into consideration this fact, it is easy to notice that the manufacturing process plans for similar products (from constructional technological point of view) made by different process engineers often differ each other. Above mentioned premises allow to purpose a thesis that it is necessary to do something with this problem. In the author' opinion it is necessary to work out, and next introduce a CAPP system in the process of technological production preparation. Currently the two methods in the process of the CAPP system designing and developing are use. The first one is the variant method and the second one is the generative method.

1.1 The Variant Method

The variant method is based on the two ideas, the idea of the products classification and the group technology. The CAPP variant system is developed in the two stages:

- preparatory stage; during this stage the following tasks are being done: working out of the products classification method, forming of the products groups, working out of the manufacturing process plans for each group of products, recording of the worked out manufacturing processes into the manufacturing database;
- operating stage; at this stage a code for a new product is being made, next the product is classified into appropriate group of products. Having read to use manufacturing processes a suitable manufacturing process plan is chosen. Next, the process plan is modified in order to adjust it to the product design description. During the adjustment process some manufacturing operations are modified, some are added or removed.

Although the variant method is a computer method it is quite similar to manual manufacturing process planning, because during the preparatory stage expert's presence is needed in order to prepare manufacturing process plans for a representative of each product group. It seems to be the biggest disadvantage of the variant method.

1.2 The Generative Method

The generative method is based on automatic synthesis of manufacturing process. In the CAPP generative system the following assumptions are made:

- product is described by means of the finite set of design features,
- for each design feature a set of alternative manufacturing operation has to be worked out.

In the figure 1 the schema of manufacturing process planning with application of the generative method is shown. In this case the manufacturing process is planned beginning from the obtained product structure description. The product description can stem from any CAD system or it can be achieved from any other sources of information such as text files etc. The product description obtained from a CAD system is usually represented by means of the design features set [5, 6, 8]. On the other hand the design feature set can be made in completely automatic way using various method of automatic design feature recognition [9] or in non–automatic way.

In the automatic feature recognition case the product model, which is first made using a CAD system built–in modelling functions, is examined according to chosen method in order to get the design description, it means the design feature set and the set of relations which exist between particular design features. In the second approach, non–automatic way, the design feature set is built basis on engineering skill and knowledge. In this process as an input information 3D models, 2D technical drawings, technical documentation are used. The worked out design feature set is next used in modelling process realised in a CAD system environment [9], and the product descriptions is most often given in text files format [2]. Having the design features set (see: the figure 1) it is possible to begin typical actions connected with manufacturing process planning: selection of manufacturing alternatives, tools selection, manufacturing process plan synthesis.

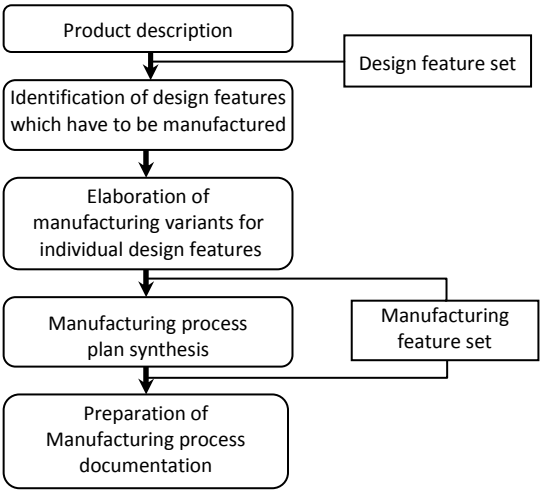


Fig. 1. The schema of the manufacturing process planning with application of the variant method

The generative CAPP systems are better than variant one. It stem from the fact that in the variant approach if the product cannot be classified in the suitable product group it is not possible to make the manufacturing process plan. In this case the variant system has to be rebuilt; it means the whole procedure of system development has to be repeated. From the other hand generative systems are built as rightful expert systems; it means that the quality of the obtained solution depends mainly on store of knowledge included in the system knowledge base and knowledge quality. In the generative systems it is possible to get solution even if the product description does not match well to the system domain. For instance if the system was designed for manufacturing process planning of axi–symmetric parts, it is also possible to plan some elements of the manufacturing processes for other group of products such as frames, levers etc. The manufacturing process planning is possible because in the generative systems the product design is represented by means of the design features set, so if it is only possible to fit suitable meta inference rule to a particular design

feature the manufacturing feature may be created. However due to lack of proper manufacturing process plans meta synthesis rules it is not possible to put in order the manufacturing features set, so it is not possible to get a complete manufacturing process plan structure.

2 The Object Design Representation

In order to develop a CAPP system development of the design representation is essential. The CAPP system oriented design representation should allow the following:

- formalisation of a product design representation in the form of a CAPP system input data. Moreover the design representation has to ensure explicit connection between design features and manufacturing techniques, it means between design features and manufacturing knowledge stored in a CAPP system manufacturing knowledge base,
- design the manufacturing process plan in the automatic way without a user interaction – an expert mode.

In the presented work the object methodology by Code and Yourdon for elaboration of the product design representation was used. In the worked out model the product design is determined by the set of instances of classes that belong to the class structure presented in the figure 2. At the current stage of the CAPP system development this class structure allows to represent product design only for the axi–symmetric products. But it is planned to widen this structure in order to make representation of the non–axi–symmetric products possible.

The base class in the class structure presented in the figure 2 is the *TProduct* class. This class represents the basic information about the product such as a product name,

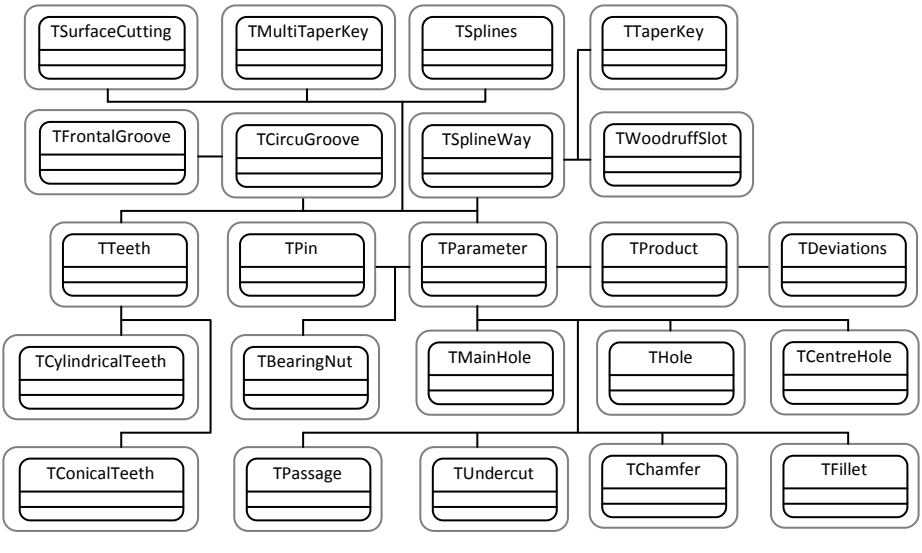


Fig. 2. The object design representation – the axi–symmetric case

product type, basic overall dimensions, product material etc. Moreover in the *TProduct* class structure special group of "manufacturing" attributes was introduced. These attributes allow to represent: semi-finished product type, production type and a type of heat treatment and additional client requirements.

Directly from this class inherits the *TParameter* and *TDeviations* class. The *TParameter* class plays a role of some kind of container which collects common methods and attributes for other classes from *TProduct* class domain so it means that this class was introduced into the object model in order to diminish model redundancy. The most important *TParameter* class attributes are the *IdDsFtSp*, the id of the superior design feature, which allows to define design feature precedence refer to its subordinate design features, and *LstDsFtSb*, the list of the subordinate design features. This information is used in a CAPP system in the process of putting the manufacturing cuts execution sequence in order at the stage of the manufacturing process structure synthesis.

In addition *TParameter* class has very important attribute called manufacturing accessibility *MnfAcc* which is used in the processes of the manufacturing strategy identification and manufacturing tools selection.

The *TDeviations* class stores in appropriate attributes lower and upper deviations values. These values can be represented both in symbolic, and numerical representations.

The remaining classes of the design object model were derived from the *TParameter* class. These classes represent particular design features distinguished at the OOA and OOD modelling stages. The one of the most important class in this model is the *TPin* class which represents the geometrical form of each design feature which belong to its domain. The *TPin* class attributes define as follows: the pin type – cylindrical or conical and pin's dimensions. The design features such as holes in the object model are represented by means of the *THole* and *TMainHole* classes. The objects that belong to the *TMainHole* class represent design features such as main holes in sleeves, disks and gear wheels products. The reaming holes which cannot be classified into the *TMainHole* class domain are represented by *THole* class. The introduction of the two classes for holes representation results from different manufacturing methods usually applied for "manufacturing" of design features such as main holes and holes which play different function in the product and have dissimilar shape. The identified design features that represent temporary fastening it means splines and multi–taper key are represented by the *TSplines* class and *TMultiTaperKey* class. In the inner structure of these classes an attribute that allows to distinguish between the design features performed on the external cylindrical surface and inner cylindrical surface was introduced. Apart of that the *TSplines* class has an attribute that distinguish the kind of the splines: external diameter fitted, internal diameter fitted and splines flank fitted. The design object model includes also the *TTeeth* class that represents design feature such as the toothed wheel rim. In order to avoid of redundancy problem the common features of the cylindrical and conical teeth are represented by the same attributes. The *TCylindricalTeeth* and *TConicalTeeth* derivative classes represent with their attributes only characteristic features of cylindrical and conical teeth. Additionally their attributes allow to distinguish between internal and external teeth and straight and slanting teeth. In the object model

which was elaborated at the stages of the object oriented analysis OOA and design OOD the group of design features such as: slots and grooves was identified. This group of features was next divided in the two subgroups of features taking into consideration the kind of functions which are performed by them. The subgroup of design features that perform the function of the shaped connections is represented by the *TSplineWay* class and its derivative classes such as the *TWoodruffSlot* and *TTaperKey*. The second subgroup represents the reaming group of grooves that perform auxiliary and technological functions in the product design. In this group of features circumferential grooves such as snap rings, toroidal sealing ring were distinguished. This group is represented by means of the *TCircumferentialGroove* class. The *TFrontalGroove* class is derivative class of the *TCircumferentialGroove* class. This class represents grooves which are performed on the frontal surface of the other design features for instance on the frontal part of the cylindrical pins etc. The design features that perform alignment and fixing functions in the design object model are represented by the *TFillet* and *TChamfer* classes. Both of them allow to distinguish between internal and external features. Manufacturing bases are represented by *TCentreHole* class. This class is used for representing all types of the centre holes such as centre holes without protecting chamfer (type A), centre holes with protecting chamfer (type B), centre holes with radius form (type C) and centre holes with screw thread. Design features performing auxiliary functions in product design relying on ensuring manufacturing accessibility to the frontal parts of pins – manufacturing undercuts are identified by the *TUndercut* class. The *TUndercut* class is used for representing all kinds of standard manufacturing undercuts both the external and internal. The design form of a bearing nut design feature which is the sum of the design forms of circumferential groove, screw thread and tangential slot is identified by means of the *TBearingNut* class. The last class in the object model is *TPassage* class which represents the design features that perform relieving functions taking into consideration fatigue strength criteria. The *TPassage* class structure allows to identify the design shapes of the following design features: radial passage, double radial passage, chamfered passage, double chamfered passage.

3 The Manufacturing Knowledge Representation

One of the most important elements of a CAPP system is a manufacturing knowledge base. In order to develop the manufacturing knowledge base it is necessary to define the term of manufacturing knowledge. In most papers by this term are meant certain information sets which allow to realise the chosen tasks from a domain of manufacturing processes design [6, 7, 8]. In this paper the term of the manufacturing knowledge was specified more precisely, so:

- store of knowledge included in the CAPP system manufacturing knowledge base depends on the scope of activities realised by the system, it means that the knowledge base should include only this kind of information which is necessary for correct solving of the decision problems. Broadening of the scope of the decision problems solved by the CAPP system is always connected with necessity of manufacturing knowledge base development;

- manufacturing knowledge has to be processed in order to record it in the CAPP system knowledge base;
- manufacturing knowledge is the set of information about the capabilities of realising manufacturing processes, and in the case of dedicated systems, about the capabilities of realising of the manufacturing process in the conditions of an enterprise taking into consideration its manufacturing recourses;
- manufacturing knowledge is the dynamic set from its nature, so it is changed under influence of continuous progress in manufacturing technology.

In the presented CAPP system in order to ensure the internal system cohesion the manufacturing knowledge, similarly to design representation method, is also represented by means of the object technique. The manufacturing knowledge object representation allows to represent and record knowledge from the scope of designing of manufacturing operations which are realised with application of the removal processes. The manufacturing knowledge is represented by means of the hierarchical class structure. Each class from this hierarchical class structure has in its internal structure the group of methods which are responsible for designing manufacturing cuts and on certain conditions manufacturing operations (in the case when it is obvious that the certain manufacturing process parts at the stage of manufacturing process structure synthesis will be for sure classified as the manufacturing operation, for instance cutting of a bar stock for the intended length). Moreover these methods allow to select cutting tools, measuring instruments and manufacturing instrumentation. This formal model of the manufacturing knowledge representation has the hybrid declarative–procedural character. The procedural character results from recording in the method inner structure the cutting tools selection procedures, measuring instrumentations and manufacturing allowances selection procedures. These selection procedures were built with applying the parameterised SQL queries to suitable tables of the CAPP system manufacturing database. The declarative character arises from recording in the particular methods content the manufacturing process structure elements design rules.

The *TProcessPlan* class is the base class in the manufacturing knowledge representation model. This class has several attributes but one of the most important attribute is *MnfObType* attribute. This attribute allow to distinguish the type of manufacturing object, so in the case of representing manufacturing operation the type of manufacturing object is *manufacturing operation* otherwise *manufacturing cut*. Taking into consideration fact that the manufacturing object can be created according to manufacturing process plan design logic for various design features it was necessary to introduce an extra class attribute *DsFtId* which stores the name of the design feature for which given manufacturing object is created. The manufacturing operation or cut content is recorded with applying of the following attributes: *RgMnfCnt*, *PrMnfCnt*, *FnMnfCnt*. These attributes store the content of rough, profiling and finishing manufacturing cuts. The *TProcessPlan* class plays also the role of the container which keeps together common attributes of remaining classes that belong to its domain.

These attributes represent for example information about manufacturing allowance values, manufacturing parameters etc. The *TMnfResources* class inherits directly from *TProcessPlan* class. This class was introduced into the manufacturing knowledge object

model in order to collect all common attributes which represent manufacturing resources such as: cutting tools, tool holders, technological instrumentation etc. The object model reaming classes inherit from the *TMnfResources* class. These classes represent particular manufacturing method, but it is necessary to remember that presented model is axi–symmetric product oriented; it means that in manufacturing knowledge base only removal manufacturing techniques are represented. The turning manufacturing technique is represented by the *TTurning* class. In the *TTurning* class structure the attributes group which represent various cutting strategies for machining of the outer cylindrical surfaces were introduced. These attributes are as follows: *RgCttSt*, *ShCttSt*, *FiCttSt*. Thanks to introducing these attributes, it is possible to design manufacturing processes according to the following cutting strategies: according to the longest cutting path, the shortest cutting path, mixed strategy. The *TTurning* class feature makes the design of the multi variant manufacturing processes possible.

The application possibility of various manufacturing recourses, for instance various insert shapes, various tool materials, insert coatings and tool holders etc. with regard to the machining type (straight turning, facing etc.), the considered manufacturing process planning stage (rough machining, profiling, finishing) caused that to the *TTurning* class inner structure the two attributes the *InsShSeq* and *HldShSeq* had to be introduced. These attributes put the selection algorithm of inserts and tool holders in order. The *TTurning* class is the base class for the two classes called adequately the *TFcTurning* (facing) and *TPrTurning* (profiling). These classes contain attributes and class methods, which take into consideration the characteristic features of the turning technique with applying of the cross–feed and the profiling machining. The centre hole machining techinque is represented by the *TCentring* class. The holes machining methods in the manufacturing knowledge representation model were recorded with applying of the following classes: the *TDrilling* class – holes drilling, *TReaming* class – holes reaming and the *TCounter* – both holes counterboring and countersinking. Above mentioned classes inherit directly from *TMnfResources* class. The milling manufacturing technique in the object model is represented by the *TMilling* class. The TMilling class is the base class for the *TPrMilling* – the profile milling, and *THbMilling* – the hobbing milling. The *TMilling* class stores with its attributes only general information about milling manufacturing technique. The characteristic features of the profile and hobbing manufacturing techniques are represented by its derivative classes. The manufacturing knowledge model was supplemented with manufacturing methods used for manufacturing of the design features such as teeth and slots. These manufacturing methods are represented by the *TSlotting* class. The grinding manufacturing method is represented by the *TGrinding* class (manufacturing of flat surfaces) which is simultaneously the base class for *TPrGrinding* class which represents the profile grinding of teeth, splines etc. The machining of the design features such as threads performed both on the external and internal surfaces is represented by the *THreading* class. The pull broaching machining method in the proposed model of manufacturing knowledge representation is described by the *TPullBroaching* class, the same machining method but used for shaped surfaces is described by the *TPrPullBroaching* class. The highly efficient method of the external and internal cylindrical surfaces machining with applying of the burnishing

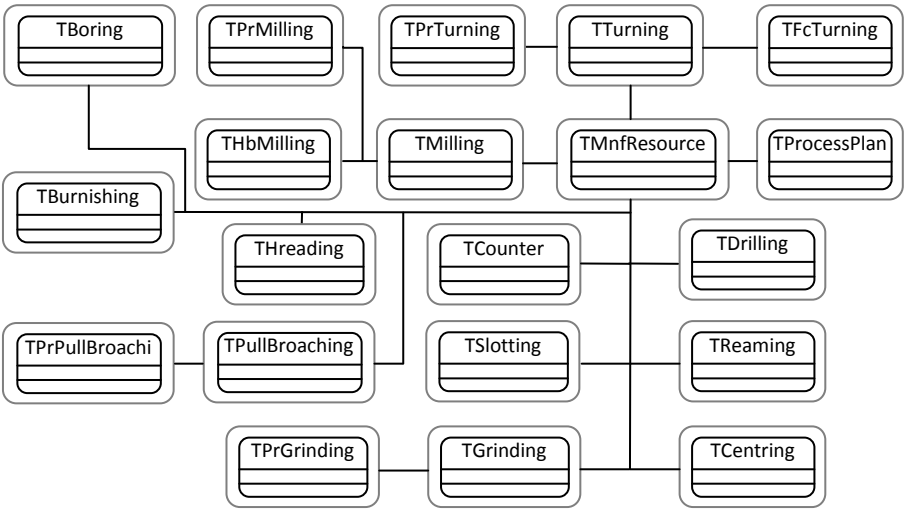


Fig. 3. The object model of manufacturing knowledge representation

manufacturing method by the *TBurnishing* class is represented. Presented in the figure 3 the object model of manufacturing knowledge representation can be widen and modified according to the CAPP system end user needs.

4 Meta Inference Rules

From the essence of manufacturing processes design results strict dependency between the product structure and course of the manufacturing process. This dependency is especially clear at the stage of manufacturing cuts selection and during the selection of manufacturing means which have to be applied in order to realise the manufacturing process. As it was mentioned in previous paragraphs an object technology was used in order to represent the product design and manufacturing process knowledge. The identified relations between these representations gave the basis for elaboration of meta inference rules. These rules were recorded in the form of production rules. The general form of the meta inference rule can be shown as follows:

IF <design feature> **THEN** <manufacturing feature>

In the meta inference rule premise part by the design feature an instance of class belonging to the class structure shown in the picture 2 is meant. On the other hand in the meta inference rule conclusion part by the manufacturing feature an instance of class belonging to the class structure shown in the picture 3 is meant. Moreover an object from rule conclusion part represents appropriate manufacturing process element structure (manufacturing operation or manufacturing cut). Here, we have to notice that the rule conclusion part can be built from many elements connected one another by means of logical operators of conjunction or alternative. So, it is possible to make more complex rules. The general shape of compound rule can be shown as follows:

IF <design feature> **THEN** <manufacturing feature_1>
OR/AND <manufacturing feature_2>
OR/AND < manufacturing feature_n>

The presence of the alternative operator in the rule conclusion part allows to record variants of manufacturing operations and cuts. It means that in the presented CAPP system it is possible to make multi–variant manufacturing processes. In the subjected system we can distinguish two the groups of meta inference rules:

- manufacturing process structure element selection rules elaborated based on identified relations between object design representation and object manufacturing knowledge representation,
- manufacturing process structure synthesis rules worked out based on frame manufacturing process and expert's knowledge.

4.1 The Manufacturing Process Structure Element Selection Rules

The manufacturing process structure element selection rules are responsible for choosing removal manufacturing cuts for particular design features. Below some examples of manufacturing process structure element selection rules are shown.

Examples of rules which can be applied for design features belonging to the domain of *THole* class.

$R1_{MPSES}$ **IF** a design feature is an instance of *THole* class **AND** a hole is "smooth" **AND** an international tolerance grade > IT11 **THEN** drilling (*TDrilling*)

$R2_{MPSES}$ **IF** a design feature is an instance of *THole* class **AND** a hole is "smooth" **AND** an international tolerance grade = IT11 **THEN** drilling (*TDrilling*) **AND** reaming (*TReaming*)

$R3_{MPSES}$ **IF** a design feature is an instance of *THole* class **AND** a hole is counterbore type having ϕ D_{ch} >= 16 with countersink **AND** an international tolerance grade < IT11 **THEN** drilling (*TDrilling*) **AND** counterboring (*TCounter*) **OR** counterbore boring (*TBoring*) **AND** countersinking (*TCounter*) **AND** reaming (*TReaming*)

Examples of rules which can be applied for design features belonging to the domain of *TMainHole* class.

$R4_{MPSES}$ **IF** a design feature is an instance of *TMainHole* class **AND** a hole is "smooth" **AND** semi-finished product is barstock **AND** an international tolerance grade > IT11 **THEN** centre hole drilling (*TCentring*) **AND** hole drilling (*TDrilling*) **OR** hole boring (*TBoring*)

$R5_{MPSES}$ **IF** a design feature is an instance of *TMainHole* class **AND** a hole is "smooth" **AND** semi-finished product is thick–walled sleeve **AND** an international tolerance grade > IT11 **THEN** hole drilling (*TDrilling*) **OR** hole boring (*TBoring*)

Examples of rules which can be applied for design features belonging to the domain of *TPin* class.

> $R6_{MPSES}$ **IF** a design feature is an instance of *TPin* class **AND** an international tolerance grade >= IT12 **THEN** rough turning (*TTurning*) **OR** rough grinding (*TGrinding*) **OR** rough milling (*TMilling*)
>
> $R7_{MPSES}$ **IF** a design feature is an instance of *TPin* class **AND** an international tolerance grade > IT8 **AND** an international tolerance grade < IT12 **THEN** rough turning (*TTurning*) **AND** (contour turning (*TPrTurning*) **OR** contour grinding (*TPrGrinding*) **OR** contour milling (*TPrMilling*))

A characteristic future above mentioned rules is that aside from selecting, these rules are able to put the sequence of manufacturing cuts worked out for certain design feature in order. The $R3_{MPSES}$ is an example of this kind of rule behaviour. This rule put the sequence of manufacturing cuts for an object which belongs to the domain of *THole* class in order. The ordered manufacturing cut sequence is made taking into consideration the demand of optimal sequence. It means that manufacturing cuts are placed in the sequence beginning form the best manufacturing cut till the worst. In the figure 4 the process of manufacturing process planning in the proposed generative CAPP is shown.

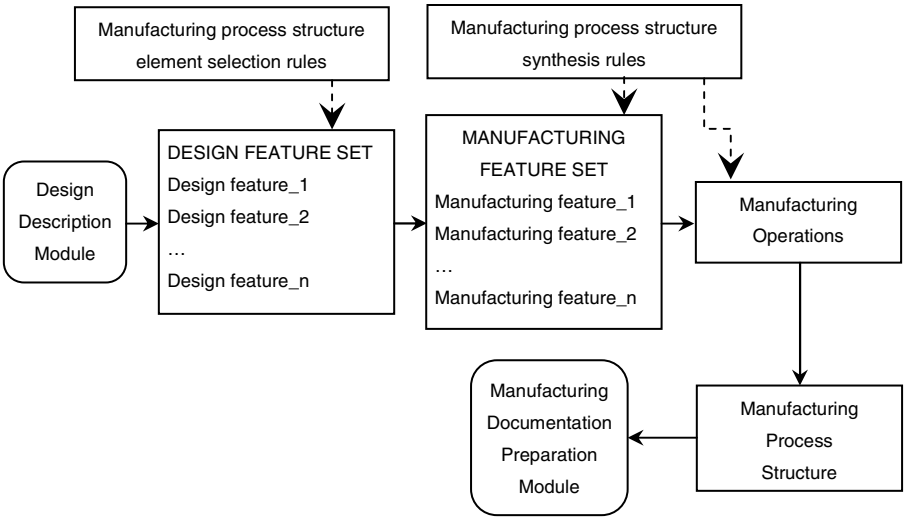


Fig. 4. The process of manufacturing process planning in the generative CAPP

5 Summary

The object methodology applied for both manufacturing knowledge representation, and the design representation allow to develop the coherent CAPP system. In the object model the product design is represented by means of object hierarchical class

structure. At the present stage of development of the proposed methodology it is possible to represent the product design only for axi–symmetric parts but the paper authors have already undertaken further research which is aimed to develop this methodology in order to represent the remaining group of products such as: frames, levers etc. Although during the manufacturing process planning the down–top approach is commonly used in the CAPP system the forward inference mechanism is applied. It means that for given set of the design features after firing appropriate rules the set of manufacturing features is created. This set is a base for preparing the complete structure of manufacturing process plan. Introduction to the design model the new design features which are characteristic for the new group of products needs to rebuild the object manufacturing knowledge model. Thanks to this it will be possible to widen the CAPP system functionality by developing the system manufacturing knowledge base. It also will need to work out the new inference rules it is manufacturing process structure element selection rules as well as manufacturing process structure synthesis rules. Currently the CAPP system works separately from a CAD system but it is planned to settle this system in the NX SIEMENS environment.

References

1. Grabowik, C., Knosala, R.: The method of knowledge representation for a CAPP system. Journal of Materials Processing Technology 133, 90–98 (2003)
2. Grabowik, C., Kalinowski, K., Monica, Z.: Integration of the CAD/CAPP/PPC. Journal of Materials Processing Technology 164-165, 1358–1368 (2005)
3. Sormaz, D., Ganduri, J.: Integration of Rule-based Process Selection with Virtual Machining for Distributed Manufacturing Planning. Springer, London (2007)
4. Krenczyk, D., Skolud, B.: Production Preparation and Order Verification Systems Integration using Method Based on Data Transformation and Data Mapping. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS (LNAI), vol. 6679, pp. 397–404. Springer, Heidelberg (2011)
5. Chen, W.L., Xie, S.Q., Zeng, F.F., Li, B.M.: A new process knowledge representation approach using parameter flow chart. Computers in Industry 66, 9–22 (2011)
6. Hassan, A., Siadat, A., Dantan, J., Martin, P.: Conceptual process planning – an improvement approach using QFD, FMEA, and ABC methods. Robotics and Computer–Integrated Manufacturing 26, 392–401 (2010)
7. Li, B.M., Xie, S.Q.: Recent development of knowledge-based systems, methods and tools for One–of–a–Kind Production. Knowledge–Based Systems 24, 1108–1119 (2011)
8. Salehi, M., Tavakkoli–Moghaddam, R.: Application of genetic algorithm to computer–aided process planning in preliminary and detailed planning. Engineering Applications of Artificial Intelligence 22, 1179–1187 (2009)
9. Hoque, A.S.M., Szecsi, T.: Designing using manufacturing feature library. Journal of Materials Processing Technology 201, 204–208 (2008)

An Experimental Study of Different Ordinal Regression Methods and Measures

P.A. Gutiérrez*, M. Pérez-Ortiz, F. Fernández-Navarro,
J. Sánchez-Monedero, and C. Hervás-Martínez

Department of Computer Science and Numerical Analysis
University of Córdoba, Spain
{pagutierrez,i82perom,i22fenaf,jsanchezm,chervas}@uco.es

Abstract. In this paper, an experimental study of different ordinal regression methods and measures is presented. The first objective is to gather the results of a considerably high number of methods, datasets and measures, since there are not many previous comparative studies of this kind in the literature. The second objective is to detect the redundancy between the evaluation measures used for ordinal regression. The results obtained present the maximum MAE (maximum of the mean absolute error of the difference between the true and the predicted ranks of the worst classified class) as a very interesting alternative for ordinal regression, being the less uncorrelated with respect to the rest of measures. Additionally, SVOREX and SVORIM are found to yield very good performance when the objective is to minimize this maximum MAE.

Keywords: Ordinal classification, ordinal regression, evaluation measures, accuracy, support vector machines, threshold models, ordinal logistic regression.

1 Introduction

The problem of *ordinal regression* is a learning problem in which the objective is to learn a rule to predict categories or labels in an ordinal scale. This problem, arisen in statistics [17], is recently receiving a lot of attention [5,6,7,9,11,21]. One can easily find the relation of this problem to the standard classification problems, since all the labels are discrete. But the difference is that there is a natural order among them. This kind of problems are also called *ranking*, *sorting* or *ordinal classification*. For example, consider the problem of classifying pictures of people in a set of four categories {`infant`,`child`,`teenager`,`adult`}. It is clear that the ranks do carry order information (`child` is definitely younger than `adult`), but ranks do not carry numerical information (`child` is not, for example, half as young as `adult`).

Compared to multi-class classification, the presence of an order relation results in two important challenges for developing models. These two challenges

* Corresponding author.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 296–307, 2012.

directly follow from the two main differences between multi-class classification and ordinal regression [4,21]. Firstly, the presence of an order relation on the classes rises a different model structure and, typically, only one global model is considered in ordinal regression, often consisting of an estimated latent variable that reflects the order on the classes [1,17]. Secondly, if an order relation on the classes can be assumed, then a performance measure that takes this order into account must be chosen, both for optimization and evaluation [21].

A formal framework for the ordinal regression problem is now introduced. Consider an input space $X \in \mathbb{R}^n$ with objects being represented by feature vectors $\mathbf{x} = (x_1, ..., x_k) \in \mathbb{R}^k$, where k is the number of features. Furthermore, let us assume that there is an outcome space $Y = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$ with ordered labels, $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_Q$. The symbol $\prec$ denotes the ordering between different ranks. A rank for the ordinal label can be defined as $\mathcal{O}(\mathcal{C}_j) = j$. The final objective in this type of problems is to find a function $f : X \to Y$ by using an i.i.d. sample $D = \{\mathbf{x}_i, y_i\}_{i=1}^N \in X \times Y$. In this context, D is distributed according to $P_{XY}^N = \prod_{i=1}^N P_{XY_i}$. Because Y contains only a finite number of labels, $P(y = \mathcal{C}_i|\mathbf{x})$ is a multinomial distribution.

Several measures can be considered for evaluating ordinal classifiers. The most common ones in Machine Learning are the Mean Absolute Error (MAE) and the Mean Zero-one Error (MZE) [7], being $MZE = 1 - Acc$, where Acc is the accuracy or correct classification rate of the classifier. However, these measures are not the best option to measure the performance in the presence of class imbalances [3]. Alternatively, it can be considered that there is an ordering for the categories, but the absolute distances among them are unknown. In that respect, in order to avoid chosing a number to represent the classes, one should only look at the order relation between "true" and "predicted" class. The use of Kendall's τ_b [13] is a step forward in that direction.

While there are many works comparing different standard classification algorithms, up to the author's knowledge there are no works comparing the performance of ordinal regressors and the difference between the measures. This paper has a double objective: it tries to gather different ordinal regression methods and datasets, in order to establish their comparative performance, but, at the same time, a study of the correlation of the different measures is done.

The paper is organized as follows: Section 2 includes a review and description of the different methods selected for this experimental study and Section 3 presents the measures used for comparison purposes. Section 4 includes the results of the study and the conclusions are drawn in Section 5.

2 Methods

Although there are some other approaches for ordinal regression (mainly based on reducing the problem to binary classification [5,11], or on simplifying it to regression [15] or cost-sensitive classification [14]), the majority of proposals can be grouped by the term *threshold methods*. These methods are based on the idea that, to model ordinal ranking problems from a regression perspective,

one can assume that some underlying real-valued outcomes exist, but they are unobservable. Consequently, two different things are estimated:

- A function $f(\mathbf{x})$ that predicts the real-valued outcomes and intends to uncover the nature of the assumed underlying outcome.
- A threshold vector $\mathbf{b} \in \mathbb{R}^{J-1}$ to represent the intervals in the range of $f(\mathbf{x})$, where $b_1 \leq b_2 \leq \ldots \leq b_{Q-1}$ (possible different scales around different ranks).

2.1 Proportional Odd Model (POM)

This is one of the first models specifically designed for ordinal regression, and it was arisen from a statistical background [17]. The model is based on the assumption of stochastic ordering of the space X, i.e. for all $\mathbf{x}_1$ and $\mathbf{x}_2$, either:

$$P(y \leq \mathcal{C}_j|\mathbf{x}_1) \geq P(y \leq \mathcal{C}_j|\mathbf{x}_2) \;\; \forall \mathcal{C}_j \in Y,$$

or:

$$P(y \leq \mathcal{C}_j|\mathbf{x}_1) \leq P(y \leq \mathcal{C}_j|\mathbf{x}_2) \;\; \forall \mathcal{C}_j \in Y. \tag{1}$$

Stochastic ordering is satisfied by a model of the form:

$$g^{-1}\left(P(y \leq \mathcal{C}_j|\mathbf{x})\right) = b_j - \mathbf{w}^T\mathbf{x},$$

where $g^{-1} : [0,1] \rightarrow (-\infty, +\infty)$ is a monotonic function often referred to as the inverse link function and b_i is the threshold defined for class $\mathcal{C}_i$. The model is naturally derived from the latent variable motivation, where $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ is a linear model. Instead of fitting a decision rule $f : X \rightarrow Y$ directly, this model defines a probability density function over the class labels for a given feature vector $\mathbf{x}$. The stochastic ordering follows from the fact that:

$$P(y \leq \mathcal{C}_j|\mathbf{x}_1) \geq P(y \leq \mathcal{C}_j|\mathbf{x}_2) \Leftrightarrow \mathbf{w}^T(\mathbf{x}_2 - \mathbf{x}_1) \geq 0,$$

and the same holds for (1). Let us assume that the ordinal response is a coarsely measured *latent* continuous variable $f(\mathbf{x})$. Thus, label $\mathcal{C}_i$ in the training set is observed if and only if $f(\mathbf{x}) \in [b_{i-1}, b_i]$, where the function f (latent utility) and $\boldsymbol{b} = (b_0, b_1, ..., b_{Q-1}, b_Q)$ are to be determined from the data. By definition, $b_0 = -\infty$ and $b_Q = +\infty$ and the real line, $f(\mathbf{x}), \mathbf{x} \in X$, is divided into Q consecutive intervals, where each interval corresponds to a category $\mathcal{C}_i$. Now, let us define a model of the latent variable, $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \epsilon$, where ϵ is the random component with zero expectation, $\mathbf{E}[\epsilon] = 0$, and distributed according to F_ϵ. Then, it follows that:

$$\begin{aligned} P(y \leq \mathcal{C}_j|\mathbf{x}) &= \sum_{q=1}^{j} P(y = \mathcal{C}_q|\mathbf{x}) = \sum_{q=1}^{j} P(f(\mathbf{x}) \in [b_{q-1}, b_q]) = P(f(\mathbf{x}) \in [-\infty, b_j]) \\ &= P(\mathbf{w}^T\mathbf{x} + \epsilon \leq b_j) = P(\epsilon \leq b_j - \mathbf{w}^T\mathbf{x}) = F_\epsilon(b_j - \mathbf{w}^T\mathbf{x}). \end{aligned}$$

If a distribution assumption F_ϵ is made for ϵ, the cumulative model is obtained by choosing, as the inverse link function g^{-1}, the inverse distribution F_ϵ^{-1} (quantile

function). Note that $F_\epsilon^{-1} : [0,1] \to (-\infty, +\infty)$ is a monotonic function. One of the most common choice for the distribution of ϵ is the logistic function (although probit, complementary log-log, negative log-log or cauchit functions could also be used). The logit is modelled using the logistic function in the following way:

$$logit(y \leq \mathcal{C}_j|\mathbf{x}) = \ln\left(odds(y \leq \mathcal{C}_j|\mathbf{x})\right) = \ln\left(\frac{P(y \leq \mathcal{C}_j|\mathbf{x})}{1 - P(y \leq \mathcal{C}_j|\mathbf{x})}\right) = \mathbf{w} \cdot \mathbf{x} + b_j.$$

In order to estimate $\mathbf{w}$ and $\boldsymbol{b}$, the vector $\mathbf{o}(\mathbf{x}_i)$ is defined, where each element $o_j(\mathbf{x}_i) = F_\epsilon^{-1}(P(y \leq \mathcal{C}_j|\mathbf{x}_i))$ is the transformed probability of categories being less than or equal to $\mathcal{C}_j$ given $\mathbf{x}_i$. This will be estimated from the sample by the transformed frequencies of this event. The estimation is achieved by using what is called the design matrix of a multivariate Generalized Linear Model (GLM). There exist methods for calculating the maximum likelihood estimate $\tilde{\mathbf{w}}$ [18] (the main difficulty is introduced by the nonlinear link function). To sum up, the POM model is considering two important assumptions about the data: a distributional assumption of the unobservable latent variable and a stochastic ordering of the space X.

2.2 Gaussian Processes for Ordinal Regression (GPOR)

GPOR [6] is a Bayesian learning algorithm, where the latent variable $f(\mathbf{x})$ is modelled using Gaussian Processes, and then all the parameters are estimated by using a Bayesian framework. The basic idea is that the values of the latent function $\{f(\mathbf{x}_i)\}$ are assumed to be the realizations of random variables indexed by their input vectors in a zero-mean Gaussian process. The ideal probability would be:

$$P(y_i|f(\mathbf{x}_i)) = \begin{cases} 1 & \text{if } b_{\mathcal{O}(y_i-1)} < f(\mathbf{x}) \leq b_{\mathcal{O}(y_i)} \\ 0 & \text{otherwise} \end{cases}.$$

The joint probability of observing the ordinal variables given the latent function is $P(D|f) = \prod_{i=1}^{N} P(y_i|f(\mathbf{x}_i))$, and the Bayes theorem is applied to write the posterior probability $P(f|D) = \frac{1}{P(D)} \prod_{i=1}^{N} P(y_i|f(\mathbf{x}_i))P(f)$.

In the presence of noise, it is explicitly assumed that the latent functions are contaminated by a Gaussian noise with zero mean and unknown variance σ^2. That it is different from the POM model, which generally considers the error distributed according to a logistic function. $P(f)$ is easily defined as a multivariate Gaussian, by using the fact that the covariance is approximated by kernels. The vector of hyperparameters $\boldsymbol{\theta}$ includes the width of the Gaussian kernels, the σ for the noise and the set of thresholds. $P(D)$ or $P(D|\boldsymbol{\theta})$ is known as the evidence for $\boldsymbol{\theta}$ and it is estimated by two different approaches in the paper: a Maximum a Posteriori approach with Laplace approximation and a Expectation Propagation with variational methods.

2.3 Support Vector Machines

The Support Vector Machine (SVM) [8] is perhaps the most common kernel learning method for statistical pattern recognition. SVM can be thought as generalized perceptrons with a kernel that computes the inner product on transformed input vectors $\phi(\mathbf{x})$, where $\phi(\mathbf{x})$ denote the feature vector $\mathbf{x}$ in a high dimensional reproducing kernel Hilbert space (RKHS) related to $\mathbf{x}$ by a specific transformation [8]. All computations are done using the reproducing kernel function only, which is defined as:

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}')$$

where $\cdot$ denotes the inner product in the RKHS.

Support Vector Machines for Binary Classification. The basic idea behind SVMs is to separate the two different classes — they are firstly defined for two classes and then extended to the multiclass case — through a hyperplane which is specified by its normal vector $\mathbf{w}$ and the bias b, $\mathbf{w} \cdot \phi(\mathbf{x}) + b = 0$, what yields the corresponding decision function:

$$f(\mathbf{x}) = y^* = \operatorname{sgn}\left(\langle \mathbf{w} \cdot \phi(\mathbf{x}) \rangle + b\right),$$

where $y^* = +1$ if $\mathbf{x}$ belongs to the corresponding class and $y^* = -1$ otherwise.

Beyond specifying non-linear discriminants by kernels, another generalization has been proposed which replaces hard margins by soft margins. This way allows to handle noise and pre-labeling errors, which often occur in practice. *Slack-variables* ξ_i are used to relax the hard-margin constraint [8]. As Vapnik [8] shows, the optimal separating hyperplane is the one which maximizes the distance between the hyperplane and the nearest points of both classes (called margin) and results in the best prediction for unseen data. In this way, the optimal separating hyperplane with maximal margin can be formulated as a Quadratic Programming (QP) optimization problem

In order to deal with the multiclass case, a "1-versus-1" approach can be considered, following the recommendations of Hsu and Lin [12].

Support Vector Machines for Ordinal Regression (SVOR). The previously defined formulation has been adapted to the ordinal regression setting, by simply defining a different threshold b_j for each class, and specifically adapting the QP problem [20]. Instead of simply deciding the class of the pattern by the sign of the projection $\mathbf{w}^T \cdot \mathbf{x}$, the corresponding real line will be split into different intervals by using a threshold vector $\mathbf{b}$. This results in parallel hyperplanes with the same $\mathbf{w}$ and different thresholds b_j. In this paper, two different implementations for this idea are considered, taken from the work of Chu and Keerthi [7]:

- *SVOR with Explicit constraints* (SVOREX) is based on defining a QP problem where the last set of constraints assuring the order between the thresholds *explicitly* appears in the optimization problem and where the slacks for the j-th parallel hyperplane are defined for all patterns of class j and $j+1$.

– *SVOR with Implicit constraints* (SVORIM) is based on redefining again the QP problem, following this principle: instead of considering only the errors from the samples of adjacent categories, samples in all the categories are allowed to contribute errors for each hyperplane. In this way, the ordinal inequalities on the thresholds are *implicitly* satisfied at the optimal solution.

2.4 Extended Binary Classification (EBC)

It is straightforward to realize that ordinal information allows ranks to be compared. For a fixed rank $\mathcal{O}(y_k) = k$, an associated question could be "is the rank of $\mathbf{x}$ greater than k?". Such a question is exactly a binary classification problem, and the rank of $\mathbf{x}$ can be determined by asking multiple questions for $k = 1, 2$, until $(Q-1)$. Frank and Hall [11] proposed to solve each binary classification problem independently and combine the binary outputs to a rank. Although their approach is simple, the generalization performance using the combination step cannot be easily analyzed. The EBC method [16] works as follows: 1) all the binary classification problems are solved jointly to obtain a single binary classifier; 2) a simple step is used to convert the binary outputs to a rank, and generalization analysis can follow.

Let us assume that $f(\mathbf{x}, k)$ is a binary classifier for all the associated questions above. A possible ranking function $r(\mathbf{x})$ based on all the binary answers $f(\mathbf{x}, k)$ is the following:

$$r(\mathbf{x}) = 1 + \sum_{k=1}^{Q-1} [\![f(\mathbf{x}, k) > 0]\!], \tag{2}$$

being $[\![\cdot]\!]$ a Boolean test which is 1 if the inner condition is true, and 0 otherwise. In summary, the EBC method is based on the following three steps:

1. Transform all training samples $(\mathbf{x}_i, y_i)$ into extended samples $(\mathbf{x}_i^{(k)}, y_i^{(k)})$:

$$\mathbf{x}_i^{(k)} = (\mathbf{x}_i, k),\ y_i^{(k)} = 2[\![k < \mathcal{O}(y_i)]\!] - 1,\ 1 \leq k \leq Q-1,$$

   but weighting these samples in the following way:

$$w_{y_i,k} = |C_{\mathcal{O}(y_i),k} - C_{\mathcal{O}(y_i),k+1}|,$$

   where C is a V-shaped cost matrix, with $C_{\mathcal{O}(y_i),k-1} \geq C_{\mathcal{O}(y_i),k}$ if $k \leq \mathcal{O}(y_i)$ and $C_{\mathcal{O}(y_i),k} \leq C_{\mathcal{O}(y_i),k+1}$ if $k \geq \mathcal{O}(y_i)$.
2. All the extended examples are then jointly learned by a binary classifier f with confidence outputs, aiming at a low weighted 0/1 loss.
3. The ranking rule of (2) is used to construct a final prediction for new samples.

This framework can be adapted to SVM, by using a threshold model to estimate $f(\mathbf{x}, k)$:

$$f(\mathbf{x}, k) = g(\mathbf{x}) - b_k,$$

where $g(\mathbf{x})$ is a non-linear function defined as $g(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x})$.

As long as the threshold vector $\boldsymbol{b}$ is ordered, i.e., $b_1 < b_2 < \cdots < b_{K-1}$, the function f is rank-monotonic. Then extended kernels need to be defined. The extended kernels of the extended examples $(\mathbf{x}, k)$ will be the original kernel plus the inner product between the extensions:

$$K((\mathbf{x}, k), (\mathbf{x}', k)) = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}') + \mathbf{e}_k \cdot \mathbf{e}_{k'},$$

where $\mathbf{E}$ is a coding matrix of $(K-1)$ rows and $\mathbf{e}_k$ is the k-th row of this matrix. Depending on the selection of $\mathbf{E}$, several SVM algorithms can be reproduced. In this paper, the coding matrix considered is $\mathbf{E} = \mathbf{I}_{K-1}$ and the cost matrix is the absolute value matrix, applied to the standard soft-margin SVM.

3 Measures

The following five measures have been used for comparison purposes:

1. *Acc*: the accuracy (*Acc*) is the rate of correctly classified patterns:

$$Acc = \frac{1}{n} \sum_{i=1}^{N} [\![y_i^* = y_i]\!],$$

   where y_i is the true rank and y_i^* is the predicted rank. *Acc* values range from 0 to 1. It represents a global performance on the classification task. Apart from not taking into account category order, it has many disadvantages, specially when unbalanced problems are considered (very different number of patterns for each class).
2. MAE: The Mean Absolute Error (MAE) is the average deviation in absolute value of the predicted class from the true class [3]:

$$MAE = \frac{1}{n} \sum_{i=1}^{N} e(\mathbf{x}_i),$$

   where $e(x_i) = |\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|$ is the distance between the true and the predicted ranks, and, then, MAE values range from 0 to $Q-1$.
3. $MMAE$: The Maximum MAE value for all the classes, i.e., $MMAE$ is the MAE value considering only the patterns from the class with higher distance from the true values to the predicted ones:

$$MMAE = \max \{MAE_j; j = 1, ..., Q\} = \max \left\{ \frac{1}{n_j} \sum_{i=1}^{n_j} e(\mathbf{x}_i); j = 1, ..., Q \right\},$$

   where MAE_j is the MAE value considering only the patterns from the *j-th* class and n_j are the number of pattern of this class. $MMAE$ values range from 0 to $Q-1$. This measure was recently proposed [10] and it is very interesting, since a low $MMAE$ represents a good ranking for all classes.

Table 1. Characteristics of the eight datasets used for the experiments: number of instances (Size), inputs (#In.), classes (#Out.) and patterns per-class (#PPC)

Dataset	Size	#In.	#Out.	#PPC
LEV	1000	4	5	(93,280,403,197,27)
SWD	1000	10	4	(32,352,399,217)
car	1728	21	4	(1210,384,69,65)
eucalyptus	736	91	5	(180,107,130,214,105)
newthyroid	215	5	3	(30,150,35)
tae	151	54	3	(49,50,52)
toy	300	2	5	(35,87,79,68,31)
winequality-red	1599	11	6	(10,53,681,638,199,19)

4. $AMAE$: The Average MAE is the mean of the MAE across classes [3]:

$$AMAE = \frac{1}{Q}\sum_{j=1}^{Q} MAE_j = \frac{1}{Q}\sum_{j=1}^{Q}\frac{1}{n_j}\sum_{i=1}^{n_j} e(\mathbf{x}_i),$$

   where $AMAE$ values range from 0 to $Q-1$.
5. τ_b: The Kendall's τ_b is a statistic used to measure the association between two measured quantities. Specifically, it is a measure of rank correlation [13]:

$$\tau_b = \frac{\sum c^*_{ij} c_{ij}}{\sqrt{\sum c^{*2}_{ij} \sum c^2_{ij}}}, i = 1, ..., N, j = 1, ..., N,$$

   where c^*_{ij} is $+1$ if y^*_i is greater than (in the ordinal scale) y^*_j, 0 if y^*_i and y^*_j are the same, and -1 if y^*_i is lower than y^*_j, and the same for c_{ij}. τ_b values range from -1 (maximum disagreement between the prediction and the true label), to 0 (no correlation between them) and to 1 (maximum agreement).

4 Experiments

The set of experiments performed in this paper include all the methods and measures considered in the previous sections and a collection of datasets from the UCI [2] and the `mldata.org` [19] repositories. Several ordinal regression works [6,7] consider real world regression datasets where the target variable is discretized into equal frequency bins. This is, in our opinion, a bit unrealistic since real world ordinal datasets (classification datasets where an order exists between the labels) are usually more complex, given the irregular distribution of patterns per class and the unobservable character of the latent variable. The characteristics of the eight datasets used for the experiments are summarized in Table 1. The synthetic toy dataset has been generated as proposed in [5].

Table 2. Average test values for each method, dataset and test measure

Measure	Method	Dataset							
		LEV	SWD	car	eucal.	newth.	tae	toy	wineq.
Acc	SVC	**0.6277**	0.5656	0.9784	*0.6529*	0.9605	0.4719	0.9591	0.6179
	POM	0.6219	0.5697	0.7309	0.1594	**0.9722**	0.5123	0.3413	0.5940
	GPOR	0.6123	**0.5777**	0.9629	**0.6855**	0.9660	0.3281	0.9538	0.6058
	SVOREX	*0.6255*	*0.5700*	*0.9877*	0.6467	0.9673	*0.5807*	*0.9818*	*0.6293*
	SVORIM	0.6179	0.5672	**0.9883**	0.6386	*0.9691*	**0.5895**	**0.9840**	**0.6303**
	EBC(SVM)	0.6236	0.5668	0.9774	0.6511	0.9685	0.5219	0.9658	0.6178
MAE	SVC	**0.4080**	0.4499	0.0216	*0.3786*	0.0395	0.5886	0.0409	0.4205
	POM	0.4112	0.4516	0.3679	2.0286	**0.0278**	0.6263	0.9178	0.4393
	GPOR	0.4219	**0.4401**	0.0376	**0.3310**	0.0340	0.8614	0.0462	0.4248
	SVOREX	*0.4096*	*0.4456*	*0.0123*	0.3920	0.0327	*0.4851*	*0.0182*	*0.4076*
	SVORIM	0.4181	0.4471	**0.0117**	0.3953	*0.0309*	**0.4605**	**0.0160**	**0.4057**
	EBC(SVM)	0.4133	0.4503	0.0226	0.3797	0.0315	0.5149	0.0342	0.4193
MMAE	SVC	**1.2411**	1.0792	0.2300	0.6339	0.1747	0.9094	0.1239	2.2750
	POM	1.3111	1.0750	2.6775	3.8074	**0.1235**	0.8712	1.7582	*2.1722*
	GPOR	1.3317	**1.0375**	0.3583	**0.5505**	0.1505	1.3859	0.1395	**2.1194**
	SVOREX	*1.2738*	1.0792	*0.1284*	*0.5807*	*0.1244*	*0.7393*	*0.0704*	2.3222
	SVORIM	1.2865	1.0708	**0.1249**	0.6039	0.1404	**0.6827**	**0.0636**	2.3211
	EBC(SVM)	1.2937	*1.0625*	0.2060	0.6465	0.1400	0.7838	0.0995	2.1750
AMAE	SVC	**0.6059**	0.6191	0.0844	*0.4093*	0.0747	0.5880	0.0437	1.1507
	POM	0.6288	*0.6058*	1.3341	1.9898	**0.0496**	0.6266	1.0763	1.0852
	GPOR	0.6544	**0.5888**	0.1446	**0.3624**	0.0623	0.8627	0.0443	**1.0647**
	SVOREX	*0.6117*	0.6140	*0.0492*	0.4110	*0.0542*	*0.4839*	*0.0178*	1.0954
	SVORIM	0.6282	0.6143	**0.0460**	0.4198	0.0550	**0.4588**	**0.0153**	1.0931
	EBC(SVM)	0.6254	0.6123	0.0773	0.4143	0.0570	0.5125	0.0327	*1.0679*
τ_b	SVC	**0.6492**	0.5244	0.9817	*0.8009*	0.9274	0.3267	0.9767	0.5071
	POM	0.6465	*0.5393*	0.1204	0.3524	0.6413	**0.6221**	0.6368	0.4262
	GPOR	0.6304	**0.5482**	0.9644	**0.8298**	0.9375	−0.0180	0.9720	0.5227
	SVOREX	*0.6479*	0.5345	*0.9897*	0.7938	0.9408	0.4453	*0.9892*	*0.5313*
	SVORIM	0.6388	0.5324	**0.9898**	0.7921	**0.9442**	*0.4819*	**0.9903**	**0.5328**
	EBC(SVM)	0.6430	0.5305	0.9768	0.7997	*0.9426*	0.4171	0.9799	0.5247

The best result is in bold face and the second one in italics.

The model performances have been evaluated by applying 30 times a holdout cross validation procedure, with $3n/4$ instances for the training set and $n/4$ instances for the generalization set. For the POM model, the `mnrfit` function of Matlab software has been used. The authors of GPOR, SVOREX, SVORIM and EBC(SVM) provide publicly available software implementations of their methods[1]. In order to establish a baseline nominal performance, the standard C-SVC with soft margin and Gaussian Kernel is considered, and the well-know `libsvm` implementation[2].

Model selection is an important issue and involves selecting the best hyper-parameter combination for all the methods compared. Model selection has been faced in this experimental study by considering a nested 10-fold validation over the training set for adjusting the width of the kernel (with values $\gamma \in \{10^{-3}, 10^{-2}, ..., 10^{3}\}$) and the cost parameter (with values $C \in \{10^{-3}, 10^{-2}, ..., 10^{3}\}$), with the same values for all the SVM methods.

The results of experiments have been included in Table 2, where the average across the 30 holdout validations is represented for each method, dataset and test

[1] GPOR (`http://www.gatsby.ucl.ac.uk/~chuwei/ordinalregression.html`), SVOREX and SVORIM (`http://www.gatsby.ucl.ac.uk/~chuwei/svor.htm`) and EBC(SVM) (`http://home.caltech.edu/~htlin/program/libsvm/`)

[2] SVC (`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`)

Table 3. Average correlation coefficients across the eight datasets ($\overline{R}$) and number of datasets where the correlations can be considered significant for $\alpha = 0.05$ ($n_{0.05}$)

$\overline{R}_{n_{0.05}}$

	GPOR					SVOREX				
	Acc	MAE	$MMAE$	$AMAE$	τ_b	Acc	MAE	$MMAE$	$AMAE$	τ_b
Acc	1.00	-0.90_7	-0.34_4	-0.64_6	0.86_8	1.00	-0.95_8	-0.37_5	-0.66_7	0.80_8
MAE	-0.90_7	1.00	0.46_5	0.77_7	-0.87_7	-0.95_8	1.00	0.43_5	0.73_7	-0.91_8
$MMAE$	-0.34_4	0.46_5	1.00	0.81_8	-0.32_4	-0.37_5	0.43_5	1.00	0.78_8	-0.41_4
$AMAE$	-0.64_6	0.77_7	0.81_8	1.00	-0.63_6	-0.66_7	0.73_7	0.78_8	1.00	-0.67_7
τ_b	0.86_8	-0.87_7	-0.32_4	-0.63_6	1.00	0.80_8	-0.91_8	-0.41_4	-0.67_7	1.00
	SVC					SVORIM				
	Acc	MAE	$MMAE$	$AMAE$	τ_b	Acc	MAE	$MMAE$	$AMAE$	τ_b
Acc	1.00	-0.94_8	-0.41_5	-0.69_7	0.81_8	1.00	-0.96_8	-0.47_5	-0.71_7	0.85_8
MAE	-0.94_8	1.00	0.42_5	0.76_7	-0.92_8	-0.96_8	1.00	0.51_6	0.77_7	-0.93_8
$MMAE$	-0.41_5	0.42_5	1.00	0.77_8	-0.39_5	-0.47_5	0.51_6	1.00	0.83_8	-0.46_4
$AMAE$	-0.69_7	0.76_7	0.77_8	1.00	-0.75_8	-0.71_7	0.77_7	0.83_8	1.00	-0.70_6
τ_b	0.81_8	-0.92_8	-0.39_5	-0.75_8	1.00	0.85_8	-0.93_8	-0.46_4	-0.70_6	1.00
	EBC(SVM)					POM				
	Acc	MAE	$MMAE$	$AMAE$	τ_b	Acc	MAE	$MMAE$	$AMAE$	τ_b
Acc	1.00	-0.95_8	-0.42_5	-0.70_7	0.80_8	1.00	-0.94_8	-0.41_5	-0.62_7	0.61_8
MAE	-0.95_8	1.00	0.47_5	0.76_7	-0.92_8	-0.94_8	1.00	0.51_5	0.75_7	-0.71_8
$MMAE$	-0.42_5	0.47_5	1.00	0.80_8	-0.44_4	-0.41_5	0.51_5	1.00	0.78_8	-0.34_4
$AMAE$	-0.70_7	0.76_7	0.80_8	1.00	-0.71_7	-0.62_7	0.75_7	0.78_8	1.00	-0.61_7
τ_b	0.80_8	-0.92_8	-0.44_4	-0.71_7	1.00	0.61_8	-0.71_8	-0.34_4	-0.61_7	1.00

The subscript is the number of datasets where the correlation is significant ($\alpha = 0.05$).

measure. A first analysis of the results reveals that some of the measures seem to be highly correlated, i.e. one measure is, approximately, a linear transformation of another. A linear correlation analysis has been performed to detect this behaviour and the results are in Table 3. A correlation matrix has been constructed for each of the eight datasets and the six methods by using the 30 test values of the five measures, and then the eight matrices has been averaged for each method. Additionally, the statistical significance of the correlations found are studied and the number of datasets where they are significant (for a significance level $\alpha = 0.05$) is included in the table. From these results, it can be concluded that the higher linear correlations are found when comparing Acc and MAE, and when comparing MAE and τ_b and the lower correlation is found between the pairs $MMAE$ and Acc, and $MMAE$ and MAE, and $MMAE$ and τ_b.

Moreover, from the matrices of correlations of Table 3, it can be concluded that the measure with the minimum correlation with respect to all the others is $MMAE$, i.e. the maximum value of MAE across all the classes. Thus, we finish the experimental study applying the Mann-Whitney U rank sum test for all pairs of algorithms when using the $MMAE$ measure in order to ascertain the statistical significance of the observed differences. The results are included in Table 4. These results include, for each algorithm, the number of algorithms statistically outperformed (wins), the number of draws (non significant differences) and the number of loses (number of algorithms that outperform the method). From the analysis of these results, the SVOREX and SVORIM methods have to be highlighted as the most competitive ones with very low number of loses and quite high number of wins, followed by EBC(SVM), GPOR and SVM. The worse method is the POM method, probably because it is a linear method, and the characteristics of the datasets involve the use of non-linear classifiers.

Table 4. Results of the Mann-Whitney U rank sum test with $\alpha = 0.05$ for $MMAE$, from a total of 40 comparisons (8 datasets by 5 compared methods)

Method	#Loses	#Draws	# Wins
SVM	13	20	7
POM	18	20	2
GPOR	16	13	11
SVOREX	5	17	18
SVORIM	4	20	16
EBC(SVM)	9	20	11

5 Conclusions

A short review of some of the different methods used for ordinal regression has been presented, together with some of the measures usually selected. A thorough experimental study comparing six methods and five measures has been done, and evaluating the correlations of the measures. This study reveals that some of the measures usually selected are highly correlated, and that the maximum MAE (maximum of the mean absolute error of the difference between true and predicted ranks of the worst ordered class) appears to be the most uncorrelated one. A set of statistical tests concludes that the best performing methods when considering maximum MAE are SVOREX and SVORIM.

Several of the consequences of this study are important for the scientific community: 1) there are not many studies comparing the performance of different ordinal regressors, and none of them considers such a high number of measures; 2) the high correlation between some of the measures is a sign that some of them are redundant; 3) the maximum MAE measure seems to be a very interesting measure for ordinal regression; and (4) SVOREX and SVORIM are very good methods when the objective is to minimize this maximum MAE.

Acknowledgments. This work has been partially subsidized by the TIN2011-22794 project of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain). Javier Sánchez-Monedero's research and Francisco Fernández-Navarro's research have been funded by the Ph. D. Student Program and the Predoctoral Program (grant reference P08-TIC-3745), respectively, both from the "Junta de Andalucía" (Spain).

References

1. Agresti, A.: Categorical Data Analysis, 2nd edn. John Wiley and Sons (2002)
2. Asuncion, A., Newman, D.: UCI machine learning repository (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
3. Baccianella, S., Esuli, A., Sebastiani, F.: Evaluation measures for ordinal regression. In: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA 2009), Pisa, Italy (December 2009)

4. Campbell, M., Donner, A.: Classification efficiency of multinomial logistic regression relative to ordinal logistic regression. Journal of the American Statistical Association 84, 587–591 (1989)
5. Cardoso, J.S., da Costa, J.F.P.: Learning to classify ordinal data: The data replication method. Journal of Machine Learning Research 8, 1393–1429 (2007)
6. Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. Journal of Machine Learning Research 6, 1019–1041 (2005)
7. Chu, W., Keerthi, S.S.: Support vector ordinal regression. Neural Computation 19(3), 792–815 (2007)
8. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
9. Crammer, K., Singer, Y.: Online ranking by projecting. Neural Computation 17(1), 145–175 (2005)
10. Cruz-Ramírez M., Hervás-Martí, C., Sánchez-Monedero, J., Gutiérrez, P.A.: A preliminary study of ordinal metrics to guide a multi-objective evolutionary algorithm. In: Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA 2011), Córdoba, Spain, pp. 1176–1181 (2011)
11. Frank, E., Hall, M.: A Simple Approach to Ordinal Classification. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, p. 145. Springer, Heidelberg (2001)
12. Hsu, C.W., Lin, C.J.: A comparison of methods for multi-class support vector machines. IEEE Transaction on Neural Networks 13(2), 415–425 (2002)
13. Kendall, M.G.: Rank Correlation Methods. Hafner Press, New York (1962)
14. Kotsiantis, S.B., Pintelas, P.E.: A Cost Sensitive Technique for Ordinal Classification Problems. In: Vouros, G.A., Panayiotopoulos, T. (eds.) SETN 2004. LNCS (LNAI), vol. 3025, pp. 220–229. Springer, Heidelberg (2004)
15. Kramer, S., Widmer, G., Pfahringer, B., de Groeve, M.: Prediction of Ordinal Classes using Regression Trees. In: Ohsuga, S., Raś, Z.W. (eds.) ISMIS 2000. LNCS (LNAI), vol. 1932, pp. 426–434. Springer, Heidelberg (2000)
16. Li, L., Lin, H.T.: Ordinal Regression by Extended Binary Classification. In: Advances in Neural Information Processing Systems, vol. 19, pp. 865–872 (2007)
17. McCullagh, P.: Regression models for ordinal data (with discussion). Journal of the Royal Statistical Society 42(2), 109–142 (1980)
18. McCullagh, P., Nelder, J.A.: Generalized Linear Models. Monographs on Statistics and Applied Probability, 2nd edn. Chapman & Hall,CRC (1989)
19. PASCAL: Pascal (Pattern Analysis, Statistical Modelling and Computational Learning) machine learning benchmarks repository (2011), http://mldata.org/
20. Shashua, A., Levin, A.: Ranking with large margin principle: Two approaches. In: Advances in Neural Information Processing Systems, vol. 15, pp. 937–944. MIT Press, Cambridge (2003)
21. Verwaeren, J., Waegeman, W., De Baets, B.: Learning partial ordinal class memberships with kernel-based proportional odds models. Computational Statistics & Data Analysis (2011) (in press) doi:10.1016/j.csda.2010.12.007

Neural Network Ensembles to Determine Growth Multi-classes in Predictive Microbiology

F. Fernández-Navarro[1], Huanhuan Chen[2], P.A. Gutiérrez[1], C. Hervás-Martínez[1], and Xin Yao[2]

[1] Department of Computer Science and Numerical Analysis, University of Cordoba, Rabanales Campus, Albert Einstein Building 3°Floor, 14071, Córdoba, Spain
[2] The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham

Abstract. This paper evaluates the performance of different ordinal regression, nominal classifiers and regression models when predicting probability growth of the *Staphylococcus Aureus* microorganism. The prediction problem has been formulated as an ordinal regression problem, where the different classes are associated to four values in an ordinal scale. The results obtained in this paper present the Negative Correlation Learning as the best tested model for this task. In addition, the use of the intrinsic ordering information of the problem is shown to improve model performance.

Keywords: Negative Correlation Learning, Neural Networks, Ordinal Regression.

1 Introduction

Growth/No-Growth models have appeared in the predictive microbiology field as an approach to determine the growth ability of microorganisms [20]. In this respect, many studies have been published in recent years for both spoilage and pathogenic microorganisms. This fact is mainly due to the need to gain more knowledge about microbial behaviour in limiting conditions that prevent growth, by using mathematical models [12]. Consequently, these mathematical models may lead to more realistic estimations of food safety risks and can provide useful quantitative data for the development of processes which lead to the production of safer food products.

The main problems in modeling microbial interface are related to its abrupt transition, i.e. the great change in the value of growth probability (p) within the very narrow range of environmental factors encountered between growth and no-growth conditions. Thus, to more properly define the interface, growth and no-growth should only be considered if all replicas (reproductions of the same conditions in environmental parameters), or at least a very high percentage of them, grow or do not grow, respectively.

One novelty of the paper is that the growth probability has been discretized into different levels of growth (classes) in order to treat it as a classification

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 308–318, 2012.

problem. Note that the exact growth probability value is not usually important for this task. In this work four observed microbial responses are obtained based on the growth probability of a microorganism, ($p = 1$ (growth), G; $0.5 \leq p < 1$ (high growth probability), GHP; $0 < p < 0.5$ (low growth probability), GLP, and $p = 0$ (no-growth), NG). The microorganism analyzed was the *Staphylococcus Aureus*. In our approach, the output of the model was the probability of pertaining to one class instead of quantifying the probability of growth.

In ordinal classification, the variable to predict is not numeric or nominal, but ordinal, so the categories have a natural order. Ordinal regression has a wide range of applications in areas where the human evaluation plays an important role, for example: psychology, medicine, information retrieval, etc., and, similarly, this method can be applied to other topics. The major problem with this type of classification is that there is not a precise notion of the distance between classes. The characteristics of the growth interface of a microorganism make that this problem can be defined as an ordinal classification problem, in which the different classes (growth intervals), can be ordered from the smallest to the largest, in increasing order. The second novel point of this paper is the use of the ordering information for obtaining better quality classifiers, and the comparison of their performance with respect to nominal classifiers.

Besides, due to the imbalanced nature of the problem (the GHP and GLP classes are clearly the minority classes), it seems natural to increase the number of replicates per condition tested. Therefore, in the preprocessing stage, the minority classes were doubled in order to improve classification performance for these classes.

In this paper, we propose a novel predictive microbiology method based on a ensemble learning technique, Negative Correlation Learning (NCL) for growth probability determination, which can generate more accurate and meaningful results. We firstly employed data preprocessing techniques (oversampling techniques). The oversampled dataset will be used for NCL to build a prediction model. We compared the classification performance of NCL with standard nominal classifiers, and ordinal regression models.

The rest of the paper is organized as follows: Section 2 shows the data preprocessing techniques applied to the predictive microbiology problem while Section 3 describes the Negative Correlation algorithm, followed by the experimental design in Section 4. Section 5 shows the results obtained and finally, our conclusions are explained in Section 6.

2 Data Preprocessing

Sampling strategies, such as over and undersampling, are extremely popular in tackling the problem of class imbalance, i.e. either the minority class is oversampled, the majority classes are undersampled, or some combination of the two is deployed (as described in [18]).

In the preprocessing stage, we have applied an oversampling process to the minority classes, in our case, the GHP and GLP classes. In this way, the initial

dataset has been modified and more synthetic samples of these classes have been created. Specifically, the number of minority class patterns (GHP and GLP patterns) was doubled.

Synthetic examples were obtained by applying the Synthetic Minority Oversampling Technique (SMOTE) algorithm [3]. SMOTE is an oversampling method where the minority class is oversampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours. Depending upon the amount of oversampling required, neighbours from the k nearest neighbours are randomly chosen. Our implementation currently uses five nearest neighbours as the maximum value of the k parameter.

3 Negative Correlation Learning for Ordinal Regression

Ordinal regression is similar to regression because the labels in either $\mathcal{Y}_r$ and $\mathbb{R}$ represent ordinal information. Nevertheless, unlike the real-valued regression labels in $\mathbb{R}$, the discrete ranks in $\mathcal{Y}_r$ do not carry metric information. That is, ordinal ranking deals with qualitative ranks while regression focuses on quantitative, real-valued outcomes. To model ordinal ranking problems from a regression perspective, it is often assumed that some underlying real-valued outcomes exist, but they are unobservable [16]. The hidden local scales "around" different ranks can be quite different, but the actual scale (metric) information is not encoded in the ranks.

Under the assumption above, each rank represents a contiguous interval on the real line. Then, ordinal ranking can be approached by the algorithm described in Fig. 1 for our problem (four classes).

Threshold Regression Algorithm:

1: Estimate a potential function $f(\mathbf{x})$ that predicts (a monotonic transform of) the real-valued outcomes.
2: Determine a threshold vector $\theta \in \mathbb{R}^{J-1}$ to represent the intervals in the range of $f(\mathbf{x})$, where $\theta_1 \leq \theta_2 \leq \ldots \leq \theta_3$

Fig. 1. Threshold Regression Algorithm

In the threshold regression algorithm, the potential function intends to uncover the nature of the assumed underlying outcome, and the threshold vector estimates the possibly different scales around different ranks. The two abstract steps of the algorithm are indeed taken by many existing ordinal ranking algorithms. For instance, in the GPOR algorithm of Chu and Ghahramani [5], the potential function $f(\mathbf{x})$ is assumed to follow a Gaussian process, and the threshold vector θ is determined by Bayesian inference with respect to some noise distribution. In the PRank algorithm of Crammer and Singer [9], the potential function $f(\mathbf{x})$ is taken to be a linear function and the pair $\langle f(\mathbf{x}), \theta \rangle$ are updated simultaneously. Some other algorithms are based on SVM, and they work on

potential functions of the form $f_v(\mathbf{x}) = \langle v, \phi(\mathbf{x}) \rangle$, where $\phi(\mathbf{x})$ maps $\mathbf{x} \in \mathbb{R}^k$ to some Hilbert space [7].

In this paper, we propose a threshold ensemble model, which is a novel instance of the threshold model. In our proposal, the thresholds of each individual in the ensemble are fixed a priori and are not modified in the learning process. Therefore, the vector θ is the same for all individuals. The vector θ is initialized to growth probability. As discussed later in this paper, we analyze the diversity among the different individuals of the ensemble. In particular we analyze the diversity in the projections of each individual. So we need that all of them project the data in a common shared space. In this space, the thresholds are the same for all the individuals of the ensemble. Moreover, we consider a standard multilayer perceptron MLP as the potential function $f(\mathbf{x})$ of each individual. Finally, the ensemble potential function is determined by simple averaging:

$$\overline{f}(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} f_i(\mathbf{x}) \tag{1}$$

where $\overline{f}(\mathbf{x})$ is the average output of the whole ensemble of M networks. In this way, the threshold ensemble model determines the class of each pattern as:

$$r_{(\overline{f}(\mathbf{x}),\theta)} = \min\{k : \overline{f}(\mathbf{x}) \leq \theta_k\} = \max\{k : \overline{f}(\mathbf{x}) > \theta_{k-1}\} \tag{2}$$

It is well known that a multi-criteria search for an ensemble that maximizes both accuracy and diversity leads to more accurate ensembles than by optimizing a single criterion. Intuitively, we expect the potential value $f(\mathbf{x})$ to be in the desired interval $(\theta_{i-1}, \theta_i]$, and we want $f(\mathbf{x})$ to be far from the boundaries (thresholds).

In our opinion, it would be very interesting to combine individuals in the ensemble so that they meet two objectives: firstly, their projections must be as far away as possible to the thresholds and secondly, their projections must be as different as possible to the projections of the remaining individuals in the ensemble. With this second objetive, we ensure that the individuals of the ensemble are accurate but their projections are different from the projections of other individuals. With the average of the projections, we intend to better estimate the real values of the latent variable. In this paper we optimize both objetives using the Negative Correlation Learning (NCL) framework [14,4].

NCL uses the following regularisation term to determine the amount of correlation in the ensemble:

$$\begin{aligned} R = p_i &= \frac{1}{N} \sum_{n=1}^{N} \left(f_i(\mathbf{x}) - \overline{f}(\mathbf{x})\right) \left(\sum_{j \neq i} f_j(\mathbf{x}) - \overline{f}(\mathbf{x})\right) \\ &= -\frac{1}{N} \sum_{n=1}^{N} \left(f_i(\mathbf{x}) - \overline{f}(\mathbf{x})\right)^2 \end{aligned}$$

Therefore, using this approach, the error of neural network i becomes:

$$e_i = \frac{1}{N}\sum_{n=1}^{N}(f_i(\mathbf{x}) - y_n)^2 + \lambda R \tag{3}$$

where λ is a weighting parameter on the regularisation term R and N the number of patterns. The λ parameter controls a trade-off between the two terms; with $\lambda = 0$ we would have an ensemble with each network trained with plain backpropagation, and as λ increases more and more emphasis would be placed on minimising the regularisation term.

4 Experimental Setup

4.1 Dataset: Staphylococcus Aureus

Staphylococcus Aureus has been recognized as an indicator of deficient food and processing hygiene and it is a major cause of food gastroenteritis worldwide. A fractional factorial design was followed in order to ascertain the growth limits of *Staphylococcus Aureus* [21] by carefully choosing a subset (fraction) of the experimental runs of a full factorial design in order to reduce experimental time and resources. The selection was based on delimiting the levels of the environmental factors studied for the growth/no-growth domain of *S. aureus* [19]. Since no growth was detected at 7.5°C or below, data were collected at 8°, 10°, 13°, 16°and 19°C, at pH levels from 4.5 to 7.5 (0.5 intervals) and at 19a_w levels (from 0.856 to 0.999 at regular intervals). The initial dataset (287 conditions) was divided into two parts: model data (training set, 146 conditions covering the extreme domain of the model) and validation data (generalization set, 141 conditions within the interpolation region of the model). Among the different conditions, there were 117 cases of G, 45 cases of GHP, 12 cases of GLP and 113 cases of NG (Table 1). The purpose of this selection was to define a dataset for model data focused on the extreme regions of the growth/no-growth domain that the boundary zones actually represent. In this study, the number of replicates per condition ($n = 30$) increased compared to other studies obtaining the growth/no-growth transition.

Table 1 shows the features of the dataset. The total number of instances or patterns in the dataset appears, as well as the number of instances in training and testing sets, the number of input variables, and the total number of instances per class. A fractional factorial design matrix form was used. This design is normally used in predictive microbiology (the fractional factorial design for *Staphylococcus Aureus* is presented in [21]). The objective of this selection was to define the training set data that actually represents the border areas in order to obtain a better fit.

4.2 Machine Learning Methods Used for Comparison Purposes and Experimental Design

For comparison purposes, different state-of-the-art methods have been included in the experimentation. These methods are the following:

Table 1. Datasets Characteristics

Dataset	#Patterns	#Training	#Test	#Inputs	#Patterns per class
S. Aureus	287	146	141	3	(117, 45, 12, 113)

– Nominal Classifiers:
  - Support Vector Machine (SVM) [22] nominal classifier is included in the experiments in order to validate our proposal contributions. Cost Support Vector Classification (SVC) available in libSVM 3.0 [2] is used as the SVM classifier implementation.
  - Other standard nominal classifiers: Other standard machine learning classifiers have been considered, given their good performance and competitiveness. They include:
    * The Logistic Model Tree (LMT) [13] classifier.
    * The C4.5 classification tree inducer [17].
    * Multi-logistic regression methods, including the MultiLogistic (MLogistic) and SimpleLogistic (SLogistic) algorithms:

– Ordinal Classifiers:
  - A Simple Approach to ordinal regression (ASA): It is straightforward to realize that ordinal information allows ranks to be compared. For a fixed rank $\mathcal{O}(y_k) = k$, an associated question could be "is the rank of x greater than k?". Such a question is exactly a binary classification problem, and the rank of x can be determined by asking multiple questions for $k = 1; 2$, until $(K-1)$. Frank and Hall [11] proposed to solve each binary classification problem independently and combine the binary outputs to a rank.
  - Support Vector Ordinal Regression (SVOR) by Chu et. al [6,8], proposes two new support vector approaches for ordinal regression. Here, multiple thresholds are optimized in order to define parallel discriminant hyperplanes for the ordinal scales. The first approach with explicit inequality constraints on the thresholds, derive the optimality conditions for the dual problem, and adapt the SMO algorithm for the solution, we will refer it to as SVOR-EX. In the second approach, the samples in all the categories are allowed to contribute errors for each threshold, thereby, there is no need of including the inequality constraints in the problem. This approach is named a SVOR with implicit constraints (SVOR-IM).
  - Gaussian Processes for Ordinal Regression (GPOR) by Chu et. al [5], presents a probabilistic kernel approach to ordinal regression based on Gaussian processes where a threshold model that generalizes the *probit* function is used as the likelihood function for ordinal variables. In addition, Chu applies the automatic relevance determination (ARD) method proposed by [15] to the GPOR model.

– Regression approaches
  - Regression neural network model (rNN): as stated in the introduction, regression models can be applied to solve the classification of ordinal data. A common technique for ordered classes is to estimate by regression any ordered scores $s_1 < \ldots < s_J$ by replacing the target class $\mathcal{C}_i$ by the score s_i. The simplest case would be setting $s_i = i$; $i = 1, \ldots, J$. A neural network with a single output was trained to estimate the scores. Furthermore, this model is particularly interesting because our ensemble model is composed by M rNN-type individuals.

Regarding different algorithms' hyper-parameters, the following procedure has been applied. For the Support Vector algorithms, i.e. SVC, SVOR-EX, and SVOR-IM, the corresponding hyper-parameters (regularization parameter, C, and width of the Gaussian functions, γ), were adjusted using a grid search with a 10-fold cross-validation, considering the following ranges: $C \in \{10^3, 10^1, \ldots, 10^{-3}\}$ and $\gamma \in \{10^3, 10^0, \ldots, 10^{-3}\}$. For GPOR-ARD no hyper-parameters were set up since the method optimizes the associated parameters itself. All the methods were configured for using the Gaussian kernel.

For the Neural Network algorithms, i.e. rNN, the corresponding hyper-parameters (number of hidden neuron, H, and number of *iterations* of the local search procedure, iterations), were adjusted using a grid search with a 5-fold cross-validation, considering the following ranges: $H \in \{5, 10, 15, 20, 30, 40\}$ and iterations $\in \{25, 50, \ldots, 500\}$. Finally, the NCL selects the λ parameter by cross validation within the range $\{0.0, 0.1, \ldots, 1.0\}$. We notice that λ could be a little greater than 1 $[\lambda \leq \frac{M}{M-1}]$ to guarantee the positive definite of Hessian matrix [1]. Since we use $M = 25$ in this paper, the up-bound of λ (1.0417) is close to 1 and we will not use the λ values which are greater than 1.

The neural network approaches (rNN and NCL) are non-deterministic methodologies. Therefore, for these algorithms, the mean and standard deviation of 30 executions are presented.

4.3 Ordinal Classification Evaluation Metrics

Four evaluation metrics have been considered which quantify the accuracy of N predicted ordinal labels for a given dataset $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N\}$, with respect to the true targets $\{y_1, y_2, \ldots, y_N\}$:

– Correct Classification Rate (C) is simply the fraction of correct predictions on individual samples:

$$C = \frac{1}{N} \sum_{i=1}^{N} I\left(\hat{y}_i = y_i\right), \tag{4}$$

where $I(\cdot)$ is the zero-one loss function and N is the number of patterns of the dataset.

– Mean Absolute Error (MAE) is the average deviation of the prediction from the true targets, i.e.:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\mathcal{O}(\hat{y}_i) - \mathcal{O}(y_i)\right|, \tag{5}$$

where $\mathcal{O}(\mathcal{C}_j) = j, 1 \leq j \leq J$, i.e. $\mathcal{O}(y_i)$ is the order of class label y_i.

These measures are aimed to evaluate two different aspects that can be taken into account when an ordinal regression problem is considered: whether the patterns are generally well classified (C) and whether the classifier tends to predict a class as close to the real class as possible (MAE).

However, in this problem it is of vital importance to know the accuracy of the most difficult classes (which in theory would be the GHP or the GLP class). As a result, we have considered two additional measures:

– Minimum Sensitivity (MS) [10]: The MS measure is defined as:

$$MS = \min\ \{S_i;\ i = 1, \ldots, J\} \tag{6}$$

where S_i is the sensitivity of the i-th class, that is, the accuracy for the class that is the worst classified.

– $MMAE$: The Maximum MAE value of all the classes. $MMAE$ is the MAE value of the class with higher distance from the true values to the predicted ones:

$$MMAE = max\ \{MAE_i;\ i = 1, \ldots, J\} \tag{7}$$

where MAE_i is the MAE value for the i-th class

5 Experimental Results

The results for the four different evaluation measures considered (CCR, MAE, MS and $MMAE$) are included in Table 2. The first conclusion is that very high accuracies are obtained, what reveals that considering the problem as a classification task can provide an accurate information of the growth interface.

From these tables, the NCL method seems to be the most competitive one from all the different alternatives considered. NCL method obtains the best results in MS and $MMAE$ and the second best results in MAE for the oversampled dataset. In the case of the original dataset, NCL achieves the best results in MAE and the second best results in CCR, MS and $MMAE$. A first analysis of the results reveals that some of the measures seem to be highly correlated. For example, from these results, it can be concluded that the higher linear correlations are found when comparing CCR and MAE. However, it seems that the $MMAE$ and MAE measures are in contrast to low levels of MAE, in a manner similar to what happened with the CCR and MS measures in nominal classification.

Table 2. Test results obtained by using the different methods evaluated

Original S. Aureus dataset					**Over-sampled S. Aureus dataset**				
Method	CCR_G	MAE_G	MS_G	$MMAE_G$	Method	CCR_G	MAE_G	MS_G	$MMAE_G$
SVC	73.04	0.4894	0.00	1.4000	SVC	68.79	0.5745	0.00	1.4000
LMT	73.04	0.4894	0.00	1.4000	LMT	68.79	0.4468	20.00	0.8000
C4.5	71.63	0.4681	0.00	1.2000	C4.5	70.92	0.4823	20.00	0.8000
MLogistic	74.46	0.4184	0.00	1.2000	MLogistic	69.50	0.4468	0.00	1.2000
SLogistic	73.04	0.4894	0.00	1.4000	SLogistic	66.66	0.5603	0.00	1.4000
ASA(C4.5)	73.04	0.4255	20.00	1.0000	ASA(C4.5)	71.63	0.4468	20.00	1.000
SVOR-EX	**79.43**	0.2553	0.00	1.000	SVOR-EX	*74.46*	**0.2836**	60.00	0.4000
SVOR-IM	**79.43**	0.2533	20.00	0.8000	SVOR-IM	68.79	0.3404	20.00	0.8000
GPOR	57.44	0.9219	0.00	1.4000	GPOR	**78.01**	0.2907	0.00	1.0000
rNN	73.75	0.2765	**69.56**	**0.3214**	rNN	73.04	0.2978	**69.54**	*0.3478*
NCL	*76.59*	**0.2482**	*56.51*	*0.4347*	NCL	73.75	*0.2907*	**69.54**	**0.3392**

The best result is in bold face and the second best result in italics

Another important conclusion is that the use of the ordering information improves the results obtained by the nominal classifiers, specially when taking into account the MAE measure: SVOR-EX improves the accuracy and MAE values of standard SVC for the problem considered; ASA(C4.5) also improves the accuracy and MAE values of standard C4.5 method both in the original dataset and in the dataset where the oversampled data have been included.

Finally, it is important to note that when the problem is unbalanced and it is useful to know the accuracy value of the minority class. It seems that the regression methods (rNN and NCL) are more accurate classifying the patterns of the minority classes than the ordinal regression methods (GPOR, SVOR-EX, and SVOR-IM). The reason is that in the regression approaches, the thresholds are fixed in the learning process while in the threshold methods (ordinal regression), the thresholds are modified during the learning process. In order to improve the accuracy of the model, the ranges of the minority classes given too narrow widths (since these classes have little importance in the overall accuracy) and this causes the accuracy of these patterns to be very small.

6 Conclusions

This paper introduced a new approach for predicting growth probability in predictive microbiology, based on a ordinal regression task rather than the usual regression or nominal classification approaches. Probability growth was discretized in four different ranges, which gather the main information needed by the experts when managing the predictive microbiology information. The results of this preliminary study show that the best performing method is the NCL, with very high accuracy and low MAE values. This paper has also shown how ordering information can still improve the performance of nominal classifiers, yielding to more accurate predictions.

Acknowledgement. This work has been partially subsidized by the TIN 2011-22794 project of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain). The research of Francisco Fernández-Navarro has been funded by the "Junta de Andalucia" Predoctoral Program, grant reference P08-TIC-3745. Huanhuan Chen and Xin Yao's work here was supported by the European Union Seventh Framework Programme under grant agreement No. 270428.

References

1. Brown, G., Wyatt, J.L., Tiňo, P.: Managing diversity in regression ensembles. Journal of Machine Learning Research 6, 1621–1650 (2005)
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
4. Chen, H., Yao, X.: Regularized negative correlation learning for neural network ensembles. IEEE Transactions on Neural Networks 20(12), 1962–1979 (2009)
5. Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. Journal of Machine Learning Research 6, 1019–1041 (2005)
6. Chu, W., Keerthi, S.S.: New approaches to support vector ordinal regression. In: ICML 2005: Proceedings of the 22nd International Conference on Machine Learning, pp. 145–152 (2005)
7. Chu, W., Keerthi, S.S.: Support vector ordinal regression. Neural Computation 19, 792–815 (2007)
8. Chu, W., Keerthi, S.S.: Support vector ordinal regression. Neural Computation 19, 792–815 (2007)
9. Crammer, K., Singer, Y.: Online ranking by projecting. Neural Computation 17, 145–175 (2005)
10. Fernández, J.C., Gutiérrez, P.A., Hervás-Martínez, C., Martínez-Estudillo, F.J.: Memetic pareto evolutionary artificial neural networks for the determination of growth limits of listeria monocytogenes. In: Proceedings of the 8th International Conference on Hybrid Intelligent Systems, Barcelona, Spain, pp. 631–636 (2008)
11. Frank, E., Hall, M.: A Simple Approach to Ordinal Classification. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 145–156. Springer, Heidelberg (2001)
12. Garcia-Gimeno, R.M., Hervas-Martinez, C., Rodriguez-Perez, R., Zurera-Cosano, G.: Modelling the growth of leuconostoc mesenteroides by artificial neural networks. International Journal of Food Microbiology 105(3), 317–332 (2005)
13. Landwehr, N., Hall, M., Frank, E.: Logistic model trees. Machine Learning 59(1-2), 161–205 (2005)
14. Liu, Y., Yao, X., Higuchi, T.: Evolutionary ensembles with negative correlation learning. IEEE Transactions on Evolutionary Computation 4(4), 380–387 (2000)
15. Mackay, D.J.C.: Bayesian methods for backpropagation networks. In: Models of Neural Networks III, ch. 6, pp. 211–254. Springer, Heidelberg (1994)
16. McCullagh, P.: Regression models for ordinal data. Journal of the Royal Statistical Society, Series B 4, 109–142 (1980)
17. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)

18. Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: Rusboost: A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans 40(1), 185–197 (2010)
19. Valero, A., Hervas, C., Garcia-Gimeno, R.M., Zurera, G.: Product unit neural network models for predicting the growth limits of listeria monocytogenes. Food Microbiology 24(5), 452–464 (2007)
20. Valero, A., Hervas, C., Garcia-Gimeno, R.M., Zurera, G.: Searching for new mathematical growth model approaches for listeria monocytogenes. Journal of Food Science 72(1), M16–M25 (2007)
21. Valero, A., Pérez-Rodríguez, F., Carrasco, E., Fuentes-Alventosa, J.M., García-Gimeno, R.M., Zurera, G.: Modelling the growth boundaries of staphylococcus aureus: Effect of temperature, P^H and water activity. International Journal of Food Microbiology 133(1-2), 186–194 (2009)
22. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1999)

Ordinal Classification Using Hybrid Artificial Neural Networks with Projection and Kernel Basis Functions

M. Dorado-Moreno, P.A. Gutiérrez, and C. Hervás-Martínez

Department of Computer Science and Numerical Analysis, University of Córdoba, Campus de Rabanales, 14071, Córdoba, Spain
{i92domom,pagutierrez,chervas}@uco.es
http://www.uco.es/ayrna

Abstract. Many real life problems require the classification of items into naturally ordered classes. These problems are traditionally handled by conventional methods intended for the classification of nominal classes, where the order relation is ignored. This paper proposes a hybrid neural network model applied to ordinal classification using a possible combination of projection functions (product unit, PU) and kernel functions (radial basis function, RBF) in the hidden layer of a feed-forward neural network. A combination of an evolutionary and a gradient-descent algorithms is adapted to this model and applied to obtain an optimal architecture, weights and node typology of the model. This combined basis function model is compared to the corresponding pure models: PU neural network, and the RBF neural network. Combined functions using projection and kernel functions are found to be better than pure basis functions for the task of ordinal classification in several datasets.

Keywords: Ordinal Classification, Projection basis functions, Kernel basis functions, Evolutionary neural networks, Evolutionary algorithm, Gradient-descent algorithm.

1 Introduction

In the real world, there are many supervised learning problems referred to as ordinal classification, where examples are labeled by an ordinal scale [24]. For instance, a teacher who rates his students using labels (A,B,C,D) that have a natural order among them (A>B>C>D). In this paper, we are selecting artificial neural networks to face this kind of problems. They are a very flexible modeling technique, whose computing power is developed using an adaptive learning process. Properties of artificial neural networks made them a common tool when successfully solving classification problems [15,17].

The objective of this paper is to adapt the hybrid model previously proposed in [14] to ordinal regression, adding a local search algorithm to result in a hybrid training method with both evolutionary and gradient-directed algorithms.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 319–330, 2012.

There are a lot of models for ordinal classification [16,23], but one of the first models specifically designed for this problem, and the one our work is based on, is the Proportional Odds Model (POM). This model is based on the assumption of stochastic ordering of the input space, and the way it works is described in [26]. In this paper, the hybrid neural network proposed in [14] is combined with the POM model to face ordinal regression.

Different types of neural networks are nowadays being used for classification purposes, including neural networks based on sigmoidal basis (SU), radial basis function (RBF) [21] and a class of multiplicative basis function, called product unit (PU) [25,28]. The combination of different basis functions in the hidden layer of a neural network has been proposed as an alternative to traditional neural networks [24]. We use RBF neurons and PU neurons according to Cohen and Intrator insights [8], based on the duality and complementary properties of projection-based functions (SU and PU) and kernel typology (RBF). These models has been also theoretically justified by Donoho [10], who demonstrated that any continuous function can be decomposed in two mutually exclusive functions, such as radial (RBF) and crest ones (SU and PU). In this way, RBF neurons contribute to a local recognition model [4], while PU neurons contribute to a global recognition one [25]. The combination of them results in a high degree of diversity because the submodels disagree one another. In a recent proposal [14], it is shown how the different combinations among these types of neurons can be achieved by an evolutionary algorithm.

In order to adjust the neural network architecture [20] which approximates the ordinal classification problem needs, training algorithms are used. One can consider gradient-directed methods such as Back-Propagation [6], which is an algorithm based on a gradient-directed search [12] resulting in a local searching. Additionally, Evolutionary Algorithms (EAs) [5,29] are an alternative, which provide a very successful platform for optimizing network weights and architecture simultaneously. Many researchers have shown that EAs perform well for global searching, because they are capable of finding promising regions in the whole search space. In this paper we will show how a hybridization of these two types of training algorithms performs very good, first performing global search with the EA and then performing a local search in the result obtained by the EA using a gradient-directed algorithm [19].

The rest of the paper is organized as follows. Section 2 discusses how the model proposal works. In section 3, the hybrid algorithm is shown. Section 4 includes the experiments: experimental design, information about the datasets, results of the experiments and statistical analysis of the results. Finally, in section 5, we present the conclusions of the paper.

2 Model

We first propose an adaption of the classical POM model [26] to artificial neural networks. Since we are using the POM model and artificial neural networks, we can say that our proposal doesnt assure monotonicity. The way the POM model

works is based on two elements: the first one is a linear layer with only one node (as seen in Fig. 1) [3], whose inputs are the non-linear transformations of a first hidden layer. The task of this node is to stamp the values into a line, to make them have an order, which allows an ordinal classification easier. After this one node linear layer, an output layer is included with one bias for each class, whose objective is to classify the patterns in the class they belong to. This classification structure corresponds to the POM model [26], which, as the majority of existing ordinal regression models, can be represented in the following general form:

$$C(\mathbf{x}) = \begin{cases} c_1, & \text{if } f(\mathbf{x}, \boldsymbol{\theta}) \leq \beta_0^1 \\ c_2, & \text{if } \beta_0^1 < f(\mathbf{x}, \boldsymbol{\theta}) \leq \beta_0^2 \\ \cdots & \\ c_J, & \text{if } f(\mathbf{x}, \boldsymbol{\theta}) > \beta_0^{J-1} \end{cases}, \tag{1}$$

where $\beta_0^1 < \beta_0^2 < \cdots < \beta_0^{J-1}$ (this will be the most important constraint in order to adapt the normal classification model to ordinal classification), J is the number of classes, $\mathbf{x}$ is the input pattern to be classified, $f(\mathbf{x}, \boldsymbol{\theta})$ is a ranking function and $\boldsymbol{\theta}$ is the vector of parameters of the model. Indeed, the analysis of (1) uncovers the general idea previously presented: patterns, $\mathbf{x}$, are projected to a real line by using the ranking function, $f(\mathbf{x}, \boldsymbol{\theta})$, and the biases or thresholds, β_0^i, are separating the ordered classes.

The POM model approximates $f(\mathbf{x}, \boldsymbol{\theta})$ by a simple linear combination of the input variables, while our model considers a non-linear basis transformation of the inputs. Let us formally define the model. For each class:

$$f_l(\mathbf{x}, \boldsymbol{\theta}, \beta_0^l) = f(\mathbf{x}, \boldsymbol{\theta}) - \beta_0^l;\ 1 \leq l \leq J-1,$$

where the projection function $f(\mathbf{x}, \boldsymbol{\theta})$ is estimated with the combination of a pair of basis functions:

$$f(\mathbf{x}, \boldsymbol{\theta}) = \beta_0 + \sum_{j=1}^{m_1} \beta_{1,j} B_{1,J}(\mathbf{x}, \mathbf{w}_{1,j}) + \sum_{j=1}^{m_2} \beta_{2,j} B_{2,j}(\mathbf{x}, \mathbf{w}_{2,j}),$$

replacing $B_{1,j}(\mathbf{x}, \mathbf{w}_{1,j})$ and $B_{2,j}(\mathbf{x}, \mathbf{w}_{2,j})$ by RBFs:

$$B_{1,j}(\mathbf{x}, \mathbf{w}_{1,j}) = B_{1,j}(\mathbf{x}, (\mathbf{c}_j, r_j)) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{2r_j^2}\right),$$

and PUs, respectively:

$$B_{2,j}(\mathbf{x}, \mathbf{w}_{2,j}) = \prod_{i=1}^{k} x_i^{w_{2,ji}}.$$

By using the POM model, this projection can be used to obtain the cumulative probabilities, cumulative odds and comulative logits of the ordinal regression in the following way [26]:

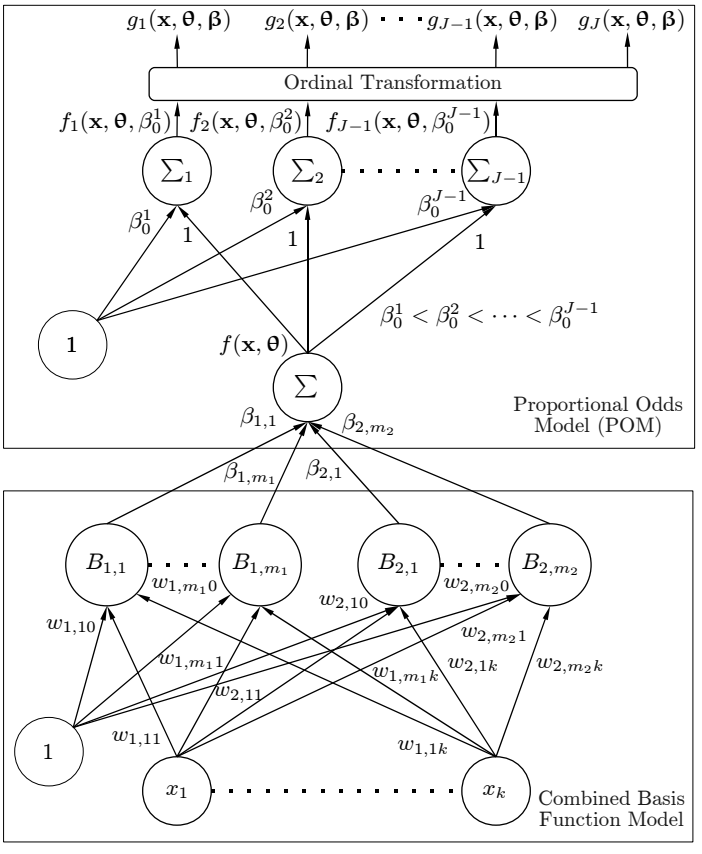


Fig. 1. Proposed hybrid model for ordinal regression

$$P(Y \leq l) = P(Y = 1) + \cdots + P(Y = l),$$

$$odds(Y \leq l) = \frac{P(Y \leq l)}{1 - P(Y \leq l)},$$

$$logit(Y \leq l) = \ln\left(\frac{P(Y \leq l)}{1 - P(Y \leq l)}\right) = f(\mathbf{x}, \boldsymbol{\theta}) - \beta_0^l,$$

$$P(Y \leq l) = \frac{1}{1 + \exp(f(\mathbf{x}, \boldsymbol{\theta}) - \beta_0^l)};\ 1 \leq l \leq J - 1,$$

$$P(Y \leq J) = 1,$$

where $P(Y = j)$ is the individual probability of a patter to belong to class j, $P(Y \leq l)$ is the probability of a pattern to belong to class 1 to l and the logit is modeled by using the ranking function, $f(\mathbf{x}, \boldsymbol{\theta})$, and the corresponding bias, β_0^l.

We can come back to $P(Y = l)$ from $P(Y \leq l)$:

$$P(Y = 1) = g_1(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) = P(Y \leq 1)$$
$$P(Y = l) = g_l(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) = P(Y \leq l) - P(Y \leq l-1),\ l = 2, \ldots, J,$$

and the final model can be expressed as:

$$g_1(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) = \frac{1}{1 + \exp(f_1(\mathbf{x}, \boldsymbol{\theta}, \beta_0^1))},$$
$$g_l(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) = \frac{1}{1 + \exp(f_l(\mathbf{x}, \boldsymbol{\theta}, \beta_0^l))} - \frac{1}{1 + \exp(f_{l-1}(\mathbf{x}, \boldsymbol{\theta}, \beta_0^{l-1}))}, l = 2, \ldots, J-1,$$
$$g_J(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) = 1 - \frac{1}{1 + \exp(f_{J-1}(\mathbf{x}, \boldsymbol{\theta}, \beta_0^{J-1}))}.$$

In order to hybridize the model, we use a percentage of RBF neurons and a percentage of PU neurons. Their training consists of firstly giving their weights ($\mathbf{w}_{1,j}$ and $\mathbf{w}_{2,j}$) random values to evolve them with the EA to get a global scope, and then, when they have better performance, applying the gradient-directed algorithm, to improve the model accuracy.

3 Algorithm

In this section, the hybrid algorithm to estimate the architecture and parameters of the model is presented. The objective of the algorithm is to design an hybrid neural network with optimal structure and weights for each ordinal classification problem. This algorithm is an extension of the neural net evolutionary programming proposed in a previous work [14]. In order to adapt the algorithm to ordinal classification, we have modified the codification of the individuals to fit the model given in section 2. The constraints, also mentioned in section 2, are $\beta_0^0 < \beta_0^1 < ... < \beta_0^{J-1}$, J being the number of classes. The algorithm is presented in Fig. 2.

To fulfill the constraints in the biases, our algorithm implements a mirror based repair of inconsistent mutation of the evolutionary part. Imagine that, after a parametric mutation, $\beta_0^0 > \beta_0^1$ so the constraints are not being fulfilled, their difference being $d = \beta_0^0 - \beta_0^1$. Our simple proposal to repair this inconsistent mutation is to move β_0^0 so that the distance to β_0^1 is the same, but fulfilling the constraints, that is, $\beta_0^0 = \beta_0^0 - 2d$.

The function used to evaluate the models and to get their fitness has also been adapted to ordinal regression. A weighted mean squared error (WeightedMSE) has been implemented, which is given by the following expression:

$$l(\boldsymbol{\beta}, \boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{J} c(y_n, l) * \left(g_l(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) - y_n^{(l)} \right)^2,$$

where several symbols must be clarified: $\mathbf{y}_n = (y_n^{(0)}, y_n^{(1)}, \ldots, y_n^{(J)})$ is a 1-of-J encoding vector of the label from pattern $\mathbf{x}_n$ (i.e. $y_n^{(j)} = 1$ if the pattern $\mathbf{x}_n$

1. Generate a random population of size NP, where each individual presents a combined basis function structure
2. Repeat until the stopping criterion is fulfilled
   2.1. Calculate the fitness (decreasing transformation of weighted MSE error) of every individual in the population and Rank the individuals respecto to their Weighted MSE error.
   2.2. Select and store best MSE individual.
   2.3. The best 10% of population individuals are replicated and substitute the worst 10% of individuals
   2.4. Apply the mutations:
      2.4.1. Parametric mutation to the best 10% of individuals.
      2.4.2. Structural mutation to the remaining 90% of individuals using a modified add node mutation in order to preserve the combined basis function structure.
   2.5. Add the best MSE individual from previous generation to the new population.
3. Select the best MSE individual in the final population.
4. Apply a gradient-directed algorithm to the best individual and consider it as a possible solution.

Fig. 2. General framework of combined basis function evolutionary programming algorithm for ordinal regression

belongs to class j, and 0 otherwise), y_n is the corresponding rank (i.e. $y_n = \arg_j(y_n^{(j)} = 1)$), $\boldsymbol{\beta} = (\beta_0^1, ..., \beta_0^{J-1})$ and $\boldsymbol{\theta}$ are the vector of biases and the vector of parameters of the ranking function, respectively, and $c(y_n, l)$ is a cost function in the following form:

$$c(y_n, l) = \begin{cases} (J/2)(J-1), & \text{if } y_n = l \\ |y_n - l|, & \text{if } y_n \neq l \end{cases} .$$

Let us illustrate the rationale behind this cost function with a problem of $J = 4$ classes. The cost of misclassifiations will be organized with the following cost matrix:

$$C_4 = \begin{pmatrix} 6\,1\,2\,3 \\ 1\,6\,1\,2 \\ 2\,1\,6\,1 \\ 3\,2\,1\,6 \end{pmatrix},$$

in such a way that the errors between individual predicted probabilities and actual ones of the MSE are differently penalized depending how far the class analyzed is from the correct class. If the class is incorrect, the penalization is the absolute value of the difference in rank. If the class is correct, the penalization is as high as the sum of the penalizations for the incorrect ones (which is $J/(2(J-1))$, i.e. the sum of the natural numbers from 1 to $J-1$).

The final step is to apply the gradient-search algorithm, which is $iRProp+$ [18]. For using this algorithm, we obtained the derivatives for the model of section 2, taking into account that we have two different types of basis functions.

4 Experiments

In order to analyze the performance of the hybrid basis function model and the corresponding hybrid training algorithm, six datasets have been tested, their characteristics shown in Table 1. The collection of datasets is taken from the UCI [1] and the `mldata.org` [27] repositories.

Table 1. Characteristics of the six datasets used for the experiments: number of instances (Size), inputs (#In.), classes (#Out.) and patterns per-class (#PPC)

Dataset	Size	#In.	#Out.	#PPC
car	1728	21	4	(1210,384,69,65)
ESL	488	4	9	(2,12,38,100,116,135,62,19,4)
LEV	1000	4	5	(93,280,403,197,27)
SWD	1000	10	4	(32,352,399,217)
tae	151	54	3	(49,50,52)
toy	300	2	5	(35,87,79,68,31)

The following two measures have been used for comparing the models:

1. CCR: The Correct Classification Rate (CCR) is the rate of correctly classified patterns:
$$CCR = \frac{1}{n}\sum_{i=1}^{N}[\![y_i^* = y_i]\!],$$
where y_i is the true label, y_i^* is the predicted label and $[\![\cdot]\!]$ is a Boolean test which is 1 if the inner condition is true. CCR values range from 0 to 1. It represents a global performance on the classification task. This measure is not taking into account category order.
2. MAE: The Mean Absolute Error (MAE) is the average deviation in absolute value of the predicted class from the true class [2]:
$$MAE = \frac{1}{n}\sum_{i=1}^{N} e(\mathbf{x}_i),$$
where $e(\mathbf{x}_i) = |y_i - y_i^*|$ is the distance between the true and the predicted ranks, and MAE values range from 0 to $J-1$. This a way of evaluating the ordering performance of the classifier.
3. K_{w}: The weighted Kappa is a modified version of the Kappa statistic calculated to allow assigning different weights to different levels of aggregation between two variables [11]:
$$K_{\mathrm{w}} = \frac{p_{o(w)} - p_{e(w)}}{1 - p_{e(w)}},$$
where
$$p_{o(w)} = \frac{1}{n}\sum_{i=1}^{J}\sum_{j=1}^{J} w_{ij} n_{ij},$$

and

$$p_{e(w)} = \frac{1}{n^2} \sum_{i=1}^{J} \sum_{j=1}^{J} w_{ij} n_{i\cdot} n_{\cdot j},$$

where n_{ij} represents the number of times the patterns are predicted by the classifier to be in class j when they really belong to class i, $n_{i\cdot} = \sum_{j=1}^{J} n_{ij}$ and $n_{\cdot j} = \sum_{i=1}^{J} n_{ij}$ for $i, j = 1, ..., J$. The weight w_{ij} quantifies the degree of discrepancy between the true (y_i) and the predicted (y_j^*) categories, and K_{w} values range from -1 to 1.

All the parameters of the algorithm are common to these six problems. The main parameters of the algorithm are: minimum and maximum number of hidden nodes ($m = 10$ and $M = 25$, respectively), population size ($N_{\mathrm{P}} = 500$), number of generations ($G_{\mathrm{E}} = 350$), number of iterations in the local search ($G_{\mathrm{LS}} = 500$) and percentage of the different types of hidden neurons ($p_{\mathrm{RBF}} = 15\%$ and $p_{\mathrm{PU}} = 85\%$). Note that the algorithm is run exactly with the same model, parameters and conditions both for pure and hybrid neural network models.

An analysis of the results obtained for both RBF and PU pure models compared with the hybrid model is performed for every dataset. We have also included two state-of-the-art classifiers:

- GPOR: Gaussian Processes for Ordinal Regression. GPOR [7] is a Bayesian learning algorithm, where the latent variable $f(\mathbf{x})$ is modelled using Gaussian Processes, and then all the parameters are estimated by using a Bayesian framework. The authors of GPOR provide publicly available software implementations of the methods[1]. For GPOR-ARD no hyper-parameters were set up since the method optimizes the associated parameters itself.
- EBC-SVM: this method applies the Extended Binary Classification (EBC) procedure to SVM [22]. The EBC method can be summarized in the following three steps. First, transform all training samples into extended samples weighting these samples by using the absolute cost matrix. Second, all the extended examples are jointly learned by a binary classifier with confidence outputs, aiming at a low weighted 0/1 loss. Last step is used to convert the binary outputs to a rank. In this way, EBC is a specific method for ordinal regression based on reformulating the problem as a binary classification problem. The hyperparameters were adjusted using a nested 10-fold cross-validation over the training set with the following values: width of the kernel,$\gamma \in \{10^{-3}, 10^{-2}, ..., 10^{3}\}$, and cost parameter, $C \in \{10^{-3}, 10^{-2}, ..., 10^{3}\}$. The evaluation measure was the test MAE error.

Both methods were configured for using the Gaussian kernel.

For the neural network methods, the experimental design was conducted using 10 random holdout procedures with 3 repetitions per holdout, $3n/4$ instances for the training set and $n/4$ instances for the generalization set (where n is the size of the dataset). For the EBC(SVM) and GPOR methods, a total of 30 random holdout procedures are performed, because of their deterministic nature.

[1] GPOR (http://www.gatsby.ucl.ac.uk/~chuwei/ordinalregression.html)

Table 2. Results of the different methods evaluated, considering test CCR, MAE and K_{w}

Dataset	Func.	$CCR(\%)$ Mean ± SD	MAE Mean ± SD	K_{w} Mean ± SD
car	Hybrid	*97.5231 ± 1.0927*	*0.0251 ± 0.0112*	**0.9884 ± 0.0065**
	PU	97.3688 ± 1.2758	0.0267 ± 0.0132	0.9431 ± 0.0267
	RBF	97.0756 ± 1.6497	0.0305 ± 0.0170	0.9367 ± 0.0348
	GPOR	96.2814 ± 0.9334	0.0376 ± 0.0098	0.9396 ± 0.0162
	EBC(SVM)	**97.7392 ± 0.8438**	**0.0226 ± 0.0084**	*0.9639 ± 0.0132*
ESL	Hybrid	**72.7595 ± 2.5891**	**0.2887 ± 0.0253**	0.6930 ± 0.0142
	PU	*72.1584 ± 4.1599*	*0.2942 ± 0.0388*	0.6505 ± 0.0514
	RBF	70.9016 ± 2.7796	0.3035 ± 0.0271	0.6340 ± 0.0346
	GPOR	71.3115 ± 3.0669	0.3008 ± 0.0346	**0.8010 ± 0.0227**
	EBC(SVM)	71.1749 ± 3.4177	0.3046 ± 0.0381	*0.7982 ± 0.0264*
LEV	Hybrid	**62.4266 ± 3.1756**	**0.4032 ± 0.0343**	0.5172 ± 0.0145
	PU	61.1733 ± 3.5606	0.4176 ± 0.0413	0.4445 ± 0.0513
	RBF	*62.4000 ± 3.0876*	*0.4034 ± 0.0331*	0.4608 ± 0.0468
	GPOR	61.2267 ± 3.0116	0.4219 ± 0.0308	*0.5459 ± 0.0374*
	EBC(SVM)	62.3600 ± 2.3477	0.4133 ± 0.0264	**0.5700 ± 0.0326**
SWD	Hybrid	*57.2133 ± 3.2419*	**0.4376 ± 0.0388**	*0.4361 ± 0.0183*
	PU	57.0800 ± 3.2583	0.4498 ± 0.0401	0.3418 ± 0.0492
	RBF	55.6667 ± 2.6967	0.4724 ± 0.0302	0.3210 ± 0.0465
	GPOR	**57.3377 ± 3.0548**	*0.4401 ± 0.0323*	**0.4390 ± 0.0427**
	EBC(SVM)	56.6800 ± 3.1111	0.4502 ± 0.0323	0.4310 ± 0.0423
tae	Hybrid	**61.4912 ± 5.6349**	**0.5035 ± 0.0723**	**0.4223 ± 0.0842**
	PU	*60.7894 ± 5.4130*	*0.5131 ± 0.0617*	*0.4117 ± 0.0808*
	RBF	54.1052 ± 7.3863	0.5526 ± 0.0854	0.3565 ± 0.1106
	GPOR	32.8070 ± 4.0729	0.8614 ± 0.1551	0.3693 ± 0.2268
	EBC(SVM)	52.1930 ± 7.3491	0.5149 ± 0.0865	0.3378 ± 0.1102
toy	Hybrid	**96.6667 ± 1.6329**	**0.0333 ± 0.0163**	*0.9666 ± 0.0211*
	PU	75.3333 ± 8.6321	0.2640 ± 0.1071	0.6758 ± 0.1146
	RBF	94.0000 ± 3.1659	0.0600 ± 0.0316	0.9217 ± 0.0414
	GPOR	95.3370 ± 2.2340	0.0462 ± 0.0223	0.9642 ± 0.0175
	EBC(SVM)	*96.5778 ± 2.4928*	*0.0342 ± 0.0212*	**0.9737 ± 0.0175**

The best result is in bold face and the second one in italics

Table 2 shows the mean test value and standard deviation of the correct classified rate (CCR) and the mean absolute error (MAE) over the 30 models obtained (10 holdout procedures ×3 repetitions or 30 holdout procedures). If we take into account the accuracy measure, the hybrid model with PUs and RBFs outperforms all the other models for 4 out of the 6 datasets. The number of datasets where the hybrid models obtain the best MAE results is 5 datasets, achieving the second best performance for the remaining one.

Although the results seem to present the hybrid model as the best performing one, it is necessary to ascertain the statistical significance of the differences observed. We follow the guidelines of Demsar [9] to achieve this purpose.

Table 3. Average rankings ($\overline{R}$) of the different methods evaluated, considering test CCR, MAE and K_{w}

	CCR		MAE		K_{w}		
Method	$\overline{R}$	p-value	$\overline{R}$	p-value	$\overline{R}$	p-value	$\alpha_{(0.1,\mathrm{Holm})}$
Hybrid	**1.33**	–	**1.25**	–	**2.00**	–	–
PU	3.33	0.006•	3.25	0.005•	3.83	0.008•	0.025
RBF	3.83	0.018•	3.75	0.006•	4.42	0.045	0.033
GPOR	3.50	0.028•	3.83	0.028•	2.42	0.648	0.050
EBC(SVM)	*3.00*	0.068•	*2.92*	0.068•	*2.33*	0.715	0.100

The best result is in bold face and the second one in italics
•: statistically significant differences for $\alpha = 10\%$

A non-parametric Friedman test [13] has been carried out with the CCR and MAE rankings of the different methods (since a previous evaluation of the C and MAE values results in rejecting the normality and the equality of variances hypothesis). The ranking is obtained for each dataset and each measure in the following way: $R = 1$ is assigned to the best method for this measures and dataset, and $R = 5$ is assigned to the worst one. The average value of this rankings are included in Table 3. The Friedman test shows that the effect of the method used for classification is statistically significant, as the confidence intervals are $C_{0.1} = (0, F_{0.1} = 2.25)$ and $C_{0.05} = (0, F_{0.05} = 2.87)$ and the F-distribution statistical values are $F^* = 3.11 \notin C_{0.05}$ for C, $F^* = 3.91 \notin C_{0.05}$ for MAE and $F^* = 4.07 \notin C_{0.05}$ for K_{w}. Consequently, the null-hypothesis stating that all algorithms perform equally in mean ranking is rejected for all measures.

Based on this rejection, the Holm post-hoc test is used to compare all classifiers to each other. Holm test is a multiple comparison procedure that considers a control algorithm and compares it with the remaining methods [9]. Holm's test adjusts the value for α in order to compensate for multiple comparison. The test is a step-up procedure that sequentially checks the hypotheses ordered by their significance. The results of the Holm test for $\alpha = 0.10$ can also be seen in Table 3, using the corresponding p and $\alpha_{(0.1,\mathrm{Holm})}$ values. From the results of this test, it can be concluded that the Hybrid methodology obtains a significantly higher ranking of CCR and MAE when compared to all the remaining methods, which justifies the proposal. The ranking of K_{w} is significantly higher than that of PU method.

5 Conclusions

A new neural network model has been proposed to face ordinal regression problems, which is mainly based on projecting the patterns into a real line and interpreting the output of the net using the Proportional Odds Model (POM) framework with a different threshold for each class, to transform them in ordered probabilities. A previously proposed nominal hybrid model, based on combining different basis functions, has been adapted to this framework, adjusting the corresponding evolutionary algorithm and including a final local search step.

The new hybrid ordinal regression neural network has been compared to the corresponding pure models which is composed of (Product Unit neural network, PU, Radial Basis Function neural network, RBF) for a total of 6 datasets, including statistical tests to evaluate the significance of the result differences. Additionally, Extended Binary Classification for Support Vector Machines [EBC(SVM)] and Gaussian Processes for Ordinal Regression (GPOR) have also been considered. Our findings reveal that, for this set of datasets, the hybrid model proposed is significantly better than the pure models and EBC(SVM) and GPOR for two of the three measures considered (accuracy and mean absolute error) and it is better (although without statistically significant differences) for the K_{w} measure.

Acknowledgments. This work has been partially subsidized by the TIN2011-22794 project of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain).

References

1. Asuncion, A., Newman, D.: UCI machine learning repository (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Baccianella, S., Esuli, A., Sebastiani, F.: Evaluation measures for ordinal regression. In: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA 2009), Pisa, Italy (December 2009)
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Information Science and Statistics, 1st edn. Springer, Heidelberg (2006)
4. Bishop, C.: Improving the generalization properties of radial basis function neural networks. Neural Computation 8, 579–581 (1991)
5. Castro, L.N., Hruschka, E.R., Campello, R.J.G.B.: An evolutionary clustering technique with local search to design rbf neural network classifiers. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 2083–2088 (2004)
6. Chauvin, Y., Rumelhart, D.E.: Backpropagation: Theory, Architectures, and Applications. Lawrence Erlbaum Associates, Inc., Mahwah (1995)
7. Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. Journal of Machine Learning Research 6, 1019–1041 (2005)
8. Cohen, S., Intrator, N.: A hybrid projection-based and radial basis function architecture: initial values and global optimisation. Pattern Analysis & Applications 5, 113–120 (2002)
9. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
10. Donoho, D.: Projection-based approximation and a duality with kernel methods. The Annals of Statistics 5, 58–106 (1989)
11. Fleiss, J.L., Cohen, J., Everitt, B.S.: Large sample standard errors of kappa and weighted kappa. Psychological Bulletin 72(5), 323–327 (1969)
12. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. Computer Journal 7, 149–154 (1964)
13. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. Annals of Mathematical Statistics 11(1), 86–92 (1940)

14. Gutiérrez, P.A., Hervás-Martínez, C., Carbonero-Ruz, M., Fernandez, J.C.: Combined projection and kernel basis functions for classification in evolutionary neural networks. Neurocomputing 27(13-15), 2731–2742 (2009)
15. Gutiérrez, P.A., Lopez-Granados, F., Peña-Barragán, J.M., Jurado-Expósito, M., Gómez-Casero, M.T., Hervás-Martínez, C.: Mapping sunflower yield as affected by *Ridolfia segetum* patches and elevation by applying evolutionary product unit neural networks to remote sensed data. Computers and Electronics in Agriculture 60(2), 122–132 (2008)
16. Hecht-Nielsen, R.: Neurocomputing. Addison-Wesley (1990)
17. Hervás-Martínez, C., Garcia-Gimeno, R.M., Martínez-Estudillo, A.C., Martínez-Estudillo, F.J., Zurera-Cosano, G.: Improving microbial growth prediction by product unit neural networks. Journal of Food Science 71(2), M31–M38 (2006)
18. Igel, C., Hüsken, M.: Empirical evaluation of the improved rprop learning algorithms. Neurocomputing 50(6), 105–123 (2003)
19. Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic search and local search in hybrid evolutionary multi-criterion optimization algorithms. IEEE Transactions on Evolutionary Computation 7(2), 204–223 (2003)
20. Koza, J.R., Rice, J.P.: Genetic generation of both the weights and architecture for a neural network. In: Proceedings of International Joint Conference on Neural Networks, vol. 2, pp. 397–404. IEEE Press, Seattle (1991)
21. Lee, S.H., Hou, C.L.: An art-based construction of RBF networks. IEEE Transactions on Neural Networks 13(6), 1308–1321 (2002)
22. Li, L., Lin, H.T.: Ordinal Regression by Extended Binary Classification. In: Advances in Neural Information Processing Systems, vol. 19, pp. 865–872 (2007)
23. Lievens, S., Baets, B.D.: Supervised ranking in the weka environment. Information Sciences 180(24), 4763–4771 (2010), http://www.sciencedirect.com/science/article/pii/S0020025510002756
24. Lippmann, R.P.: Pattern classification using neural networks. IEEE Transactions on Neural Networks 27, 47–64 (1989)
25. Martínez-Estudillo, A.C., Martínez-Estudillo, F.J., Hervás-Martínez, C., García, N.: Evolutionary product unit based neural networks for regression. Neural Networks 19(4), 477–486 (2006)
26. McCullagh, P.: Regression models for ordinal data (with discussion). Journal of the Royal Statistical Society 42(2), 109–142 (1980)
27. PASCAL: Pascal (pattern analysis, statistical modelling and computational learning) machine learning benchmarks repository (2011), http://mldata.org/
28. Schmitt, M.: On the complexity of computing and learning with multiplicative neural networks. Neural Computation 14, 241–301 (2001)
29. Yao, X.: Evolving artificial neural networks. Proceedings of the IEEE 87(9) (1999)

A Genetic Programming Approach for Solving the Linear Ordering Problem

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. The linear ordering problem (LOP) consists in rearranging the rows and columns of a given square matrix such that the sum of the super-diagonal entries is as large as possible. The LOP has a significant number of important practical applications. In this paper we describe an efficient genetic programming based algorithm, designed to find high quality solutions for LOP. The computational results obtained for two sets of benchmark instances indicate that our proposed heuristic is competitive to previous methods for solving the LOP.

Keywords: Linear ordering problem, heuristics, evolutionary computation, genetic programming.

1 Introduction

The linear ordering problem (LOP) is a classical combinatorial optimization problem which appeared in the literature under various names: the maximum acyclic subdigraph problem, the maximum consistent arc set, or the median ordering problem. LOP has been subject of research for at least 53 years, beginning with the solutions described by Chenry and Watanabe [4] in 1958.

Formally, the LOP can be stated as follows: given a square matrix $A = [a_{ij}]_{n\times n}$, we are interested in finding a permutation of the rows and columns such that the sum of the entries in the upper triangle is maximized.

The problem plays an important role due to the large number of applications in various areas including economics, scheduling, social sciences, electrical engineering, machine learning, ranking in sports tournaments, etc. For more information on the problem and its applications we refer to [3,13].

Example. Given the following 4×4 - matrix:

$$A = \begin{pmatrix} 0 & 12 & 10 & 6 \\ 22 & 0 & 13 & 15 \\ 23 & 23 & 0 & 14 \\ 31 & 27 & 20 & 0 \end{pmatrix}$$

we observe that the sum of the entries in the upper triangle is 70.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 331–338, 2012.

Performing the following permutations: $L < 1,4 >, L < 2,3 >, C < 1,4 >$ and $C < 2,3 >$ (where $L < i,j >$ means an interchange of lines i and j and $C < i,j >$ means an interchange of the columns i and j), we get the matrix:

$$\tilde{A} = \begin{pmatrix} 0 & 20 & 27 & 31 \\ 15 & 0 & 23 & 23 \\ 14 & 13 & 0 & 22 \\ 6 & 10 & 12 & 0 \end{pmatrix}$$

having the sum of the entries in the upper triangle 146, which is the optimum permutation of the rows and columns in order to maximize the sum of the entries in the upper triangle.

It has been proved that the linear ordering problem is NP-hard [8] and therefore the problem cannot be solved exactly in realistic time for practical problem instances.

The difficulty of obtaining optimal solutions for practical problem instances has led to the development of several heuristic and metaheuristic algorithms: the first solution methods have been proposed by Chenry and Watanabe [4], a multi-start method was described by Chanas and Kobylanski [2] is based on an insertion mechanism that searches for the best position to insert a sector in the partial ordering under construction,a Tabu Search introduced by Laguna et al. [12] that includes an intensification phase using short-term memory based on a tabu search criterion, a diversification process through a long-term memory and an additional intensification process that applies a path relinking strategy based on elite solutions. Campos et al. [1] developed a heuristic approach based on scatter search, Schiavinotto and Stutzle [17] proposed a memetic algorithm obtained by combining a genetic algorithm with a single local search on an insertion neighborhood and Garcia et al. [7] described a heuristic algorithm based on variable neighborhood search (VNS) and as well a hybrid method that combines the VNS with a short-term tabu search for improved outcomes. Pintea et al. [15] proposed a hybrid heuristic based on ant algorithms in order to solve the triangulation problem for Input-Output tables and Chira et al. [5] investigated ant models for solving LOP The memetic algorithm described by Schiavinotto and Stutzle [17] is the state-of-the-art heuristic for solving LOP.

A benchmark library and a comprehensive survey on heuristic algorithms for solving the linear ordering problem was given by Marti et al. [13,14].

The aim of this paper is to describe an improved heuristic based on genetic programming for solving the LOP. Our algorithmic approach is tested against state-of-the-art heuristic algorithms on two sets benchmark instances from the literature LOLIB and MBLB. As will be shown in the computational experiments section, the proposed approach provides high quality solutions.

2 The Genetic Program for Solving the LOP

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 [11] 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.

As in the case of GA, in GP two models of evolution are used: the standard generational algorithm that creates new offspring from the members of an old population using genetic operators, see Cobb and Grefenstette [6], and the incremental/steady state algorithm where typically there is only one new member inserted into the new population at any time, see Whitley and Kauth [18].

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 LOP based on genetic programming.

2.1 Genetic Representation

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

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

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 the given order.

Given the following matrix:

$$A = \begin{pmatrix} 0 & 12 & 10 & 6 \\ 22 & 0 & 13 & 15 \\ 23 & 23 & 0 & 14 \\ 31 & 27 & 20 & 0 \end{pmatrix} \tag{2}$$

and the individual $I = (L < 1,4 >, C < 1,4 >)$, the matrix undergoes the following permutations:

1. the lines 1 and 4 are interchanged and results the following matrix:

$$A_1 = \begin{pmatrix} 31 & 27 & 20 & 0 \\ 22 & 0 & 13 & 15 \\ 23 & 23 & 0 & 14 \\ 0 & 12 & 10 & 6 \end{pmatrix}, \tag{3}$$

2. the columns 1 and 4 are interchanged and the result is:

$$A_2 = \begin{pmatrix} 0 & 27 & 20 & 31 \\ 15 & 0 & 13 & 22 \\ 14 & 23 & 0 & 23 \\ 6 & 12 & 10 & 0 \end{pmatrix}. \tag{4}$$

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 sum of super-diagonal entries of the matrix resulted after a program is applied to the original matrix.

2.4 Genetic Operators

We considered two operations for modifying structures in genetic programming: crossover and mutation. 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. Offspring are produced from two parent solutions using the following crossover procedure: 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:

$$P_1 = (M_1^1 M_2^1 | M_3^1 M_4^1), \tag{5}$$
$$P_2 = (M_1^2 M_2^2 | M_3^2 M_4^2 M_5^2), \tag{6}$$

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:

$$O_1 = (M_1^1 M_2^1 | M_3^2 M_4^2 M_5^2), \tag{7}$$
$$O_2 = (M_1^2 M_2^2 | M_3^1 M_4^1). \tag{8}$$

Mutation. Mutation is another important feature of genetic programming. If at the beginning the evolutionary algorithms used only one mutation operator in producing the next generation, it was shown that each optimization problem may require different mutation operators in order to obtain best results, for more details we refer to [10]. We use in our GP two random mutation operators chosen with the same probability:

1. exchange of two moves: two alleles randomly selected are swapped.
2. 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.

The probability of applications of the genetic operators is called operator rate. 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%.

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

In this section we present computational results to assess the effectiveness of our proposed genetic programming heuristic for solving the LOP.

We conducted our experiments on two sets of instances: MBLB and LOLIB. MBLB comprises 30 instances: 5 instances of size 100, 10 of size 150, 10 of size 200 and 5 of size 250 and are generated randomly, with the matrix entries generated according to a uniform distribution, and then certain number of zeros are added. LOLIB comprises 49 real world instances, of which 30 are of size n = 44, 5 of size n = 50, 11 of size n = 56, and 3 of size n = 60.

The testing machine was an Intel Dual-Core 1,6 GHz and 2 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 and running times of our genetic programming based heuristic with some of the best algorithms from the literature: the Chanas and Kobylanski heuristic algorithm ($\mathcal{CK}$) [2], the local search heuristics proposed by Schiavinotto and Stutzle [16] that consists of the following neighborhoods: insert neighborhood ($\mathcal{N}_{\mathcal{I}}$), interchange neighborhood ($\mathcal{N}_{\mathcal{X}}$) and the concatenations of $\mathcal{N}_{\mathcal{I}}$ with $\mathcal{N}_{\mathcal{X}}$ ($\mathcal{N}_{\mathcal{I}}$+$\mathcal{N}_{\mathcal{X}}$), $\mathcal{N}_{\mathcal{X}}$ with $\mathcal{N}_{\mathcal{I}}$ ($\mathcal{N}_{\mathcal{X}}$+$\mathcal{N}_{\mathcal{I}}$), $\mathcal{N}_{\mathcal{I}}$ with $\mathcal{CK}$ ($\mathcal{N}_{\mathcal{I}} + \mathcal{CK}$), $\mathcal{CK}$ with $\mathcal{N}_{\mathcal{I}}$ ($\mathcal{CK} + \mathcal{N}_{\mathcal{I}}$), the memetic algorithm ($\mathcal{MA}$) described by Schiavinotto and Stutzle [17], Tabu Search [12] ($\mathcal{TS}$) and the scatter search ($\mathcal{SS}$) developed by Campos et al. [6].

For the results of obtained using our genetic programming $\mathcal{GP}$ algorithm, we report the average deviation computed over all the results obtained for 100 runs in the case of both sets of instances LOLIB and MBLB.

The columns in the table are as follows: the first column contains the set of used instances, the second column represent the heuristic algorithm, then the Average deviation (%) gives the percentage deviation from the known optimal solutions, # optima the number of optimal solutions found and finally in the last column we present our obtained running times and the corresponding running times reported in the original papers.

Analyzing the computational results reported in Table 1, it is clear that our GP based heuristic algorithm is competitive to the earlier proposed heuristic approaches for solving the LOP. The performance of our proposed method is clearly superior with respect to the number of optimal solutions: 49 as in the case of the memetic algorithm [17] for LOLIB instances and 15 in the case of MBLB instances.

Table 1. Comparison of our GP algorithm to other algorithms from the literature LOLIB and MBLB instances

	Heuristic alg.	Average dev. (%)	# optima	Running time (s)
LOLIB	$\mathcal{CK}$	0.2403	38	0.0205
	$\mathcal{TS}$	0.04	30	0.33
	$\mathcal{SS}$	0.01	42	3.82
	$\mathcal{N}_{\mathcal{I}}$	0.1842	42	0.1802
	$\mathcal{N}_{\mathcal{I}} + \mathcal{CK}$	0.1819	44	0.1881
	$\mathcal{CK} + \mathcal{N}_{\mathcal{I}}$	0.236	40	0.042
	$\mathcal{MA}$	0.00	**49**	0.00176
	$\mathcal{GP}$	0.24	**49**	15.24
MBLB	$\mathcal{CK}$	0.0209	12	0.22
	$\mathcal{N}_{\mathcal{I}}$	0.0195	10	10.04
	$\mathcal{N}_{\mathcal{X}}$	0.387	0	9.35
	$\mathcal{N}_{\mathcal{X}} + \mathcal{N}_{\mathcal{I}}$	0.0182	11	23.33
	$\mathcal{N}_{\mathcal{I}} + \mathcal{N}_{\mathcal{X}}$	0.0191	14	11.44
	$\mathcal{N}_{\mathcal{I}} + \mathcal{CK}$	0.0169	10	9.94
	$\mathcal{CK} + \mathcal{N}_{\mathcal{I}}$	0.0197	14	0.9
	$\mathcal{GP}$	0.4	**15**	12.21

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 linear ordering problem based on genetic programming. The computational experiments show that our approach behaves well in comparison with the best heuristics proposed for LOP in terms of solution quality. In the future we plan to improve the performance of our heuristic algorithm by considering other or additional genetic operators and the running times by developing a parallel implementation strategy of the algorithm.

Acknowledgments. This research is supported by Grant PN II TE 113/2011, New hybrid metaheuristics for solving complex network design problems, funded by CNCS Romania.

References

1. Campos, V., Glover, F., Laguna, M., Marti, R.: An Experimental Evaluation of a Scatter Search for the Linear Ordering Problem. Journal of Global Optimization 21, 397–414 (2001)

2. Chanas, S., Kobylanski, P.: A new heuristic algorithm solving the linear ordering problem. Computational Optimization and Applications 6, 191–205 (1996)
3. Charon, I., Hudry, O.: A survey on the linear ordering problem for weighted or unweighted tournaments. 4OR: A Quarterly. Journal of Operations Research 5(1), 5–60 (2007)
4. Chenery, H.B., Watanabe, T.: International Comparisons of the Structure of Production. Econometrica 26(4), 487–521 (1958)
5. Chira, C., Pintea, C.M., Crisan, G.C., Dumitrescu, D.: Solving the Linear Ordering Problem using Ant Models. In: Proc. of GECCO 2009, pp. 1803–1804. ACM (2009)
6. Cobb, H., Grefenstette, J.: GA for tracking changing environments. In: Proc. of the Int. Conf. on Genetic Algorithms (1993)
7. Garcia, C.G., Perez-Brito, D., Campos, V., Marti, R.: Variable neighborhood search for the linear ordering problem. Computers and Operations Research 33, 3549–3565 (2006)
8. 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)
9. Huang, G., Lim, A.: Designing a Hybrid Genetic Algorithm for the Linear Ordering Problem. In: Cantu-Paz, E., et al. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 1053–1064. Springer, Heidelberg (2003)
10. Hong, T.-P., Wang, H.-S., Chen, W.-C.: Simultaneously Applying Multiple Mutation Operators in Genetic Algorithms. Journal of Heuristics 6(4), 439–455 (2000)
11. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge (1992)
12. Laguna, M., Marti, R., Campos, V.: Intensification and diversification with elite tabu search solutions for linear ordering problem. Computers and Operations Research 26, 1217–1230 (1998)
13. Marti, R., Reinelt, G.: The Linear Ordering Problem. Exact and Heuristic Methods in Combinatorial Optimization. Applied Mathematical Sciences, vol. 175. Springer, Heidelberg (2011)
14. Marti, R., Reinelt, G., Duarte, A.: A Benchmark Library and a Comparison of Heuristic Methods for the Linear Ordering Problem. Computational Optimization and Applications (to appear)
15. Pintea, C.M., Crisan, G.C., Chira, C., Dumitrescu, D.: A Hybrid Ant-Based Approach to the Economic Triangulation Problem for Input-Output Tables. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 376–383. Springer, Heidelberg (2009)
16. Schiavinotto, T., Stützle, T.: Search Space Analysis of the Linear Ordering Problem. In: Raidl, G.R., Cagnoni, S., Cardalda, J.J.R., Corne, D.W., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E., Meyer, J.-A., Middendorf, M. (eds.) EvoIASP 2003, EvoWorkshops 2003, EvoSTIM 2003, EvoROB/EvoRobot 2003, EvoCOP 2003, EvoBIO 2003, and EvoMUSART 2003. LNCS, vol. 2611, pp. 322–333. Springer, Heidelberg (2003)
17. Schiavinotto, T., Stutzle, T.: The linear ordering problem: Instances, search space analysis and algorithms. Journal of Mathematical Modelling and Algorithms 3, 367–402 (2004)
18. Whitley, D., Kauth, J.: GENITOR: A different genetic algorithm. In: Proc. of the Rocky Mountain Conf. on Artificial Intelligence. Denver (1988)

Comparison of Fuzzy Functions for Low Quality Data GAP Algorithms

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

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

Data gathered by this way are called Low Quality Data (LQD). Thus, uncertainty representation tools are needed for using in learning models with this kind of data.

This work presents a method to represent the uncertainty and an approach for learning white box Equation Based Models (EBM). The proficiency of the representations with different noise levels and fitness functions typology is compared.

The numerical results show that the use of the described objectives improves the proficiency of the algorithms. It has been also proved that each meta-heuristic determines the typology of fitness function.

Keywords: Low Quality Data, Simulated Annealing, Genetic Programming Algorithm, Equation Based Model.

1 Introduction

It is well known that some kind of commercial sensors don't keep an exact relationship with the measure. Furthermore, these sensors have several differences: like hysteresis, non-linear ranges, etc.

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. Data gathered by this way are called Low Quality Data (LQD) and they have a high level of vagueness in relation with the measure.

The necessity of LQD learning algorithms is well known in literature [5]: the higher vagueness and uncertainty the worse the proficiency of the Non-LQD learning algorithms.

The fuzzy rule based systems (FRBS) are used like models to represent the uncertainty of one process. Nevertheless, the complexity of the data sets must

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 339–349, 2012.

be analyzed to choose the best FRBS. Several measures are analyzed in [9] using the same learning algorithm. These measure could be extended to LQD.

In our opinion, one of the most successful researches in soft computing dealing with LQD is detailed in [3]. This work shows the mathematical basis for learning uncertainty aware of genetic fuzzy systems -both classifiers and models-. The LQD is assumed as fuzzy data, where each α-cut represents an interval value for each data.

Finally, it is worth pointing out that the fitness functions to train classifiers and models with LQD are also fuzzy valued functions. Hence the learning algorithms should be adapted to such fitness functions [11].

These ideas have been applied in some tools using realistic data based on real world problems [13,17].

Present study resumes an uncertainty representation proposal based on white box equation based models (EBM). This proposal uses the variant multi-objective of the meta-heuristic Simulated Annealing (MOSA), [14], to evolve hybrid genetic structures of genetic algorithm and genetic programming (GAP), [6] which represent the uncertainty model. Two different methods to evaluate the models are compared, one based on α-cut values and the second based on the lower-upper limits of the error, both lacking of total order relationship. The behavior of the learning of the models taking as input well known functions with different levels of noise is analyzed.

The remainder of this manuscript is as follows. Firstly, in the next section the representation of the EBM vagueness evolving vague GAP structures by MOSA algorithm is presented. Later, in section 3 the two evaluation methods are analyzed. In section 4, data sets used in the comparison are presented and numerical results are commented. Finally, some conclusions and future work are outlined.

2 Representation of the Vagueness in GAP Models

Although vagueness can be represented in the data set [17], that chance supposes higher computational cost and complexity in the learning process.

This study proposes an uncertainty representation considering discrete data. Thus, EB like relationship between an output variable and different input variables are used to prove out proposal, figure 1.

Genetic Programming (GP) is typically used in problems where the learning of equations is dealt with [7]. While GP evolves equations that are self-contained (that is, the equation contains the values of the constants that eventually appear in the formulae) these are usually avoided to reduce the diversity and the time consumption in finding good solutions. Instead of including the constants in the equation, GP is generally hybridized with Genetic Algorithm (GA): the GP learning for evolving the structure of a model and the GA for evolving its numeric constants [14]. In the latter case, when a constant is to be used in an equation, the node that represents the constant links to a valid position in the vector of constants of the individual. GAP is the choice in this study (see figure 1).

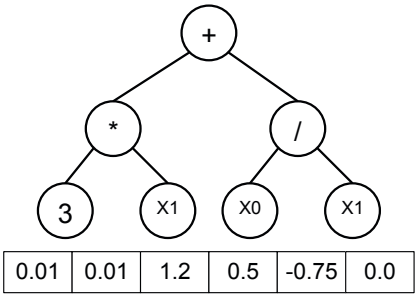


Fig. 1. The representation of an EB model. The tree on the left corresponds with y = 0.5 * x1 + x0/x1. Each model contains a tree of nodes for storing the structure of the equation and a vector of numerical constants.

Vagueness representation in EB has been rarely studied in literature. Learning EB models with LQD using GP hybridized with GA (hereinafter, GAP) has been barely studied [10,12] with the inclusion of vagueness constants in the EB.

Conversely, in [16] a proposal about the vagueness representation in constant vectors is presented.

For each imprecise variable we assign two constants C^- and C^+, which are to be evolved in the learning process, whenever $X1$ is evaluated for the example j, a fuzzy number with a triangular fuzzy membership defined through the three following values $[d_1^j - C^-, d_1^j, d_1^j + C^+]$ is returned.

It is interesting to mention, that if symmetrical membership functions are adopted, then only one constant per imprecise variable is needed.

Present study applies this second vagueness EB representation. Consequently, the number of free constants in the equations is reduced.

This proposal of vagueness representation does the evaluation of a model with crisp data and the output is not a crisp data but a fuzzy number. Thus, the number of available constants for an equation is reduced.

2.1 Representation of the GAP Individual

As in GAP models, the equation representation consists of a grammar driver nodes tree, each internal node corresponds with a valid operator, and the leaf nodes correspond with a variable index or a constant index (see ver Figura 2).

The number of constants to represent the uncertainty in each variable could be 0 if no imprecise variable is assumed, 1 if symmetrical triangular membership functions are used or 2 if asymmetrical triangular membership functions are assumed. The number of imprecise variables is predefined.

The first group of constants in the vector are reserved for uncertainty management and the remainder are free to be indexed in the equations.

As it is stated in [1], the random initialization of individuals improves the generation methods with a-priori information, so the former is the method used here.

Trees have been generated driven by following grammar rules:

EXP $\mapsto$ MonoOperator | BinaryOperator | Constant | Variable
MonoOperator $\mapsto$ (sin EXP) | (cos EXP) | (**delay** EXP)
BinaryOperator $\mapsto$ (+ EXP EXP) | (− EXP EXP) | (∗ EXP EXP) |
(/ EXP EXP) | (min EXP EXP) | (max EXP EXP)

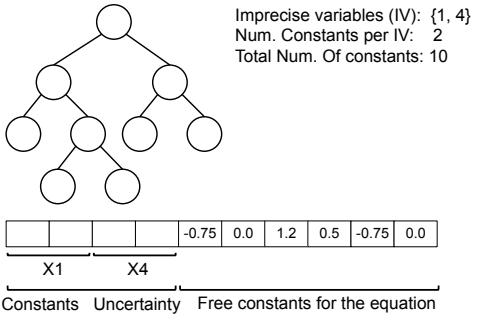


Fig. 2. The representation of an individual: the nodes tree to represent the equation, the uncertainty specification and the constants vector. The latter is divided into the uncertainty constants and the constants that remain in the equation.

2.2 Genetic Operators

Evolving EB individuals involves using four genetic operators: two from GP evolution (the GP crossover and mutation) and two from GA (GA crossover and mutation). The GP operators introduce variability in the structure of the model, that is, 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 operations.

The GP crossover is defined as follows: two parents are chosen to be crossed using binary tournament; then, for each one an index is randomly generated (the index is in the range from 1 to the number of nodes in the tree); finally, the nodes at the index positions are interchanged. The GP mutation operator modifies the type of one operator to another operator type of the same arity.

A node indexing an equation constant mutates varying its index among the valid indexes for equation constants in the constants vector of the individual.

The GA crossover is a classical two point crossover (the constants vector is then divided in three parts: the initial, the central and the ending parts) that interchanges the constants vector of both individuals. Here individual are selected by binary tournament selection. On the other hand, for each constant

in the constants vector of an individual, the GA mutation operator evaluates whether to mutate or not, according to a predefined probability, and if so the constant is assigned with a random value in the also predefined range of the constant values.

3 Fitness Fuction

Given the uncertainty representation, the EB evaluation taking input discrete data generates a fuzzy number.

Thus, if the fitness function is based on the square error or mean square error (MSE) the output is a fuzzy number. If the goal is finding the best fitness model, a partial or total order relationship must be for the list of fitnesses.

Although it's possible to set a total order relationship for fuzzy numbers [8], it's necessary a-priori knowledge of the problem. For each problem the right order relationship must be selected to obtain optimal results. By contrast, this study shows the comparison of two evaluation methods, to set a partial order relationship.

The first order relationship is presented in [16], called αMSE. Here one of the goals of the fitness function is the minimization of the mean square error. This measure is simplified using an α-cut, taking the interval range like error for one datum and the global error like the sum of the interval square error for whole dataset. To determine if a value is lower (or higher) than another an interval comparison must be done and the intersection must be empty, otherwise the intervals are not comparable between them.

Also, this proposal includes two more supplementary goals: the minimization of *mean interval width* (WIDTH) of the output model for an α-cut, and the maximization of the *percentage the covered dataset examples* (PCT), i.e., those intervals which include the output model for an $\alpha - cut$.

In [11] is described the second order relationship, which uses the mean square error for each example in the data set called *boundedMSE*.

When the square error is a fuzzy number and it includes the origin, the fuzzy number is folded over the positive axe.

Later, the membership functions areas below and under the maxima membership value are integrated getting the low and high limits for the square error. To obtain the interval value each example of the data set is added. The interval comparison is the same than the one for the WIDTH variable.

4 Experiments and Numerical Results

4.1 Experiments

To illustrate the difference between the different fitness functions analyzed here several data sets have been generated with a selection of well-known formula taken from the literature which are described in the next section. Each dataset has been generated with three different levels of imprecision in the input data:

0%, 1% and 5% in the range variables. The same output reference calculated for the non-noise input is used for all the data set.

All the experiments include the evaluation of the EB learning like described in [16], using meta-heuristic MOSA, [14]. In present work one only objective (minimization of the square error), two objectives (maximization of PCT is added) and three objectives (MSE, PCT and WIDTH) are analysed.

The last fitness proposal (square error, PCT and WIDTH) with the three levels of imprecision is compared with the proposal presented in [11]. The remaining fitness function with one and two goals is not included in this comparison because of an early convergence of the method due to the avid behavior of the MSE measure with the uncertainty representation proposal.

Although, enough experiments were done and probe the problem of early convergence but those are not included here because of lack of space.

Following parameters were used for each experiment. Only imprecise variables were considered, using one constant to represent the uncertainty per variable with a range in variation of 1% of the variable value.

Models with a maximum of 10 constants and absolute value of 10.0 were used. The maximum depth is 7, while the maximum number of nodes is set to 15. The size of population is 100 individuals with 1000 epochs; and the crossover and mutation probability is 0.75, and 0.25 for GP and GA. MOSA uses $\Delta = 0.916229$, $T_{init} = 1$ and $T_{fin} = 0$

4.2 Standard Data Sets

Several data sets have been generated with a selection of well-known formula taken from the GP's literature. Each data set includes 5 input variables and 250 examples. The values of the input variables were randomly generated in a specific range for each function. These are the selected formula:

T2: This formulae was taken from [2], and the data set output is $f(x_1) = \sin(x_1) \cdot x_2 + 5$, con $x_i \in [-5, 5]$, $\forall i = 1, \ldots, 5$. The first variable is the only one used for the output calculus.

P3: A polynomial function taken from [4], its data set output is $f(x_1) = 1.5 + 24.3 \cdot x_1^2 - 15 \cdot x_1^3 + 3.2 \cdot x_1^4$, with $x_i \in [-0, 1]$, $\forall i = 1, \ldots, 5$. The first variable is the only one used for the output calculus.

F2: The Ackley function is referred in [15]. It output data set is $f(\{x_i\}) = 20 - 20 \cdot \exp(-0.2 \cdot \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} x_i^2}) + \exp(1) - \exp(\frac{1}{N} \cdot \sum_{i=1}^{N} \cos(2 \cdot \pi \cdot x_i))$, with $x_i \in [-100, 100]$, $\forall i = 1, \ldots, 5$. The first two variables are used for the output calculus.

F5: The scaled Schwefel function [15] is $f(\{x_i\}) = 4.189828872724339 \cdot N - 0.01 \cdot \sum_{i=1}^{N} (x_i \cdot \sin(\sqrt{|x_i|})$, with $x_i \in [-500, 500]$, $\forall i = 1, \ldots, 5$. Again, first two variables are used for the output calculus.

F8: The discontinuous Schaffer function [15] is $f(\{x_i\}) = \sum_{i=1}^{N} F(y_i, y_{(i+1)\%N})$, with $x_i \in [-500, 500]$, taking $y_i = x_i$ if $|x_i| \leq 0.5$, $y_i = round(2 \cdot x_i)/2$ else, % is the module operator, $F(x, y) = 0.5 + \frac{\sin^2(\sqrt{x^2+y^2}]) - 0.5}{1+0.001 \cdot \sqrt{x^2+y^2}}$ and $\forall i = 1, \ldots, 5$. The output depends on the two first input variables.

F9: The exapanded and rotated Schaffer function [15] , is $f(\{x_i\}) = \sum_{i=1}^{N} F(x_i, x_{(i+1)\%N})$, con $x_i \in [-500, 500]$, % es el operador mdulo, $F(x, y) = 0.5 + \frac{\sin^2(\sqrt{x^2+y^2}]) - 0.5}{1+0.001 \cdot \sqrt{x^2+y^2}}$ y $\forall i = 1, \ldots, 5$. The output depends on the two first input variables.

4.3 Numerical Results

For each data set and noise level ten independent runs have been carried out, selecting in each one the individual with lower central point square error. Numerical results using evaluation with α-MSE [16] and one objective MSE and two objectives (MSE and PCT) are shown in table 1.

Table 1. Results with one objective (MSE) and two objectives ((PSE+PCT) using α-MSE evaluation. DS is the data set, NL is the noise level, MDN y AVE are the median and the average of the central points for 10 runs.

Data set		α-MSE only MSE		α-MSE MSE + PCT	
DS	**NL**	**MDN**	**AVE**	**MDN**	**AVE**
T2	0.00	0.0017	0.0004	2.2444	0.0555
	0.01	0.0014	0.0004	5.0000	0.0717
	0.05	0.0009	0.0005	0.0667	0.0001
P3	0.00	0.0006	0.0004	3.6442	0.0015
	0.01	0.0018	0.0008	5.0000	0.0023
	0.05	0.0033	0.0006	5.0000	0.0284
F2	0.00	0.0212	0.0113	1.2402	0.0476
	0.01	0.0236	0.0126	5.0000	1.5854
	0.05	0.0078	0.0036	4.9459	1.2482
F5	0.00	0.0823	0.0276	5.0000	5.0000
	0.01	0.0233	0.0174	5.0000	5.0000
	0.05	0.0354	0.0221	5.0000	0.4953
F8	0.00	0.0026	0.0006	5.0000	0.6258
	0.01	0.0234	0.0017	5.0000	0.3170
	0.05	0.0016	0.0012	5.0000	0.3557
F9	0.00	0.0298	0.0020	5.0000	0.2589
	0.01	0.0516	0.0002	1.6353	0.1562
	0.05	0.1141	0.0104	5.0000	0.1011

Results with the three objectives: MST, PCT and mean width of the interval (WIDTH) are shown in table 2. The first two columns use evaluation with α-MSE [16] while the last two columns were evaluated with *bounded*-MSE [11].

Figure 3 shows the comparative depicted boxplot about the EB learning using α-MSE evaluation against an increasing of the noise data.

Table 2. Results with three objectives Resultados (MSE+PCT+WID) using α-MSE and *bounded-MSE* evaluation. DS is the data set, NL is the noise level, MDN y AVE are the median and the average of the central points for 10 runs.

		α-MSE		bounded-MSE	
Data set		MSE+PCT+WID		MSE+PCT+WID	
DS	**NR**	**MDN**	**AVE**	**MDN**	**AVE**
T2	0.00	0.0084	0.0002	0.0209	0.0120
	0.01	0.0018	0.0002	0.0025	0.0001
	0.05	0.0052	0.0002	0.0401	0.0055
P3	0.00	0.0013	0.0004	0.0172	0.0000
	0.01	0.0209	0.0007	0.0178	0.0076
	0.05	0.0022	0.0009	0.0073	0.0053
F2	0.00	0.0168	0.0056	0.1430	0.1031
	0.01	0.0094	0.0030	0.2679	0.1167
	0.05	0.0131	0.0062	0.1140	0.0919
F5	0.00	0.0092	0.0012	0.9504	0.5703
	0.01	0.0049	0.0002	0.4204	0.4243
	0.05	0.0483	0.0036	0.4196	0.0960
F8	0.00	0.0367	0.0003	0.2603	0.0010
	0.01	0.0025	0.0001	0.5675	0.0331
	0.05	0.0413	0.0002	0.1941	0.0131
F9	0.00	0.0346	0.0002	0.0083	0.0044
	0.01	0.0269	0.0001	0.0181	0.0082
	0.05	0.0683	0.0007	0.0089	0.0070

Finally it can be concluded that the proposal with two objectives (MSE+PCT) reduces the performance of the learning algorithm.

By contrast, the fitness with only one objective MSE doesn't include the data set examples.

On the other hand, the proposal with three objectives generates EB with lower imprecision level, getting higher number of examples. It can be concluded, that the three objectives are required to get a good EB evolution using our uncertainty representation.

Also, results in table1 show that the α-MSE evaluation method is more suitable for MOSA algorithm that *bounded*-MSE. By contrast, *bounded*-MSE needs higher number of iterations and individuals in the population to obtain good proficiency.

Results that show this statement can't be included here, due to lack of space. Also, this second method is more sensible to the kind of problem, so it can be concluded that *bounded*-MSE is more suitable for meta-heuristics like NSGA-II than MOSA-like.

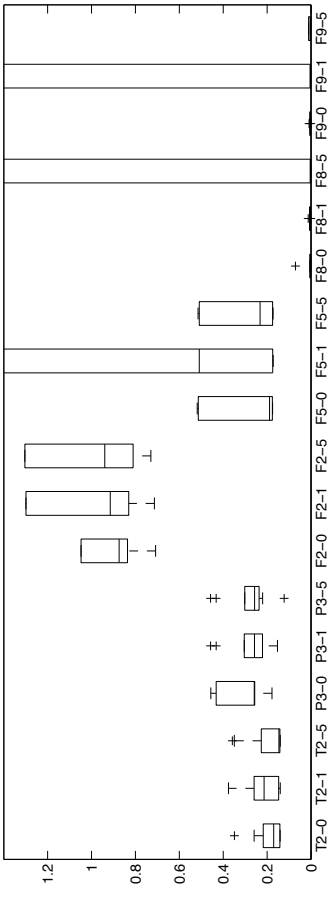


Fig. 3. Boxplot using α-MSE evaluation with different data set and noise levels (0, 1, and 5 corresponds with 0.00%, 0.01% and 0.05%

5 Conclusions and Future Work

This study proposes an uncertainty representation schema for EB to be learnt using evolutive techniques. The influence of the noise in data sets as different evaluation methods for the models have been compared. The results conclude that the three objectives based fitness improve the proficiency of the algorithms. Likewise, it was proved that each meta-heuristic determines the typology of fitness function for the models.

In future work it could be considered these three aspects: the inclusion of vague constants in our uncertainty representation proposal, the unification

of the uncertainty for all the models and the study of the population diversity in order to spread the search space.

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. Berzosa, A., Villar, J.R., Sedano, J., García-Tamargo, M., de la Cal, E.: An Study of the Tree Generation Algorithms in Equation Based Model Learning with Low Quality Data. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS, vol. 6679, pp. 84–91. Springer, Heidelberg (2011)
2. Brameier, M., Banzhaf, W.: Explicit Control of Diversity and Effective Variation Distance in Linear Genetic Programming. In: Foster, J.A., Lutton, E., Miller, J., Ryan, C., Tettamanzi, A.G.B. (eds.) EuroGP 2002. LNCS, vol. 2278, pp. 37–49. Springer, Heidelberg (2002)
3. Couso, I., Sánchez, L.: Higher order models for fuzzy random variables. Fuzzy Sets Syst. 159, 237–258 (2008)
4. Ekárt, A., Németh, S.Z.: A Metric for Genetic Programs and Fitness Sharing. In: Poli, R., Banzhaf, W., Langdon, W.B., Miller, J., Nordin, P., Fogarty, T.C. (eds.) EuroGP 2000. LNCS, vol. 1802, pp. 259–270. Springer, Heidelberg (2000)
5. Folleco, A., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: Identifying learners robust to low quality data. Informatica (Slovenia) 33(3), 245–259 (2009)
6. Howard, L., D' Angelo, D.: The ga-p: a genetic algorithm and genetic programming hybrid. IEEE Expert 10, 11–15 (1995)
7. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
8. Lee-Kwang, H., Lee, J.-H.: Method for ranking fuzzy numbers and its application to decision-making. IEEE Transactions on Fuzzy Systems 7(6), 677–685 (1999)
9. Luengo, J., Herrera, F.: Domains of competence of fuzzy rule based classification systems with data complexity measures: A case of study using a fuzzy hybrid genetic based machine learning method. Fuzzy Sets and Systems 161(1), 3–19 (2010); Special section: New Trends on Pattern Recognition with Fuzzy Models
10. Sánchez, L.: Interval-valued gap algorithms. IEEE Transactions on Evolutionary Computation 4, 64–72 (2000)
11. Sánchez, L., Couso, I., Casillas, J.: Genetic learning of fuzzy rules based on low quality data. Fuzzy Sets and Systems 160(17), 2524–2552 (2009)
12. Sánchez, L., Couso, I., Corrales, J.A.: Combining gp operators with sa seach to evolve fuzzy rule classifiers. Information Sciences 136, 175–192 (2001)
13. 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)
14. Sánchez, L., Villar, J.R.: Obtaining transparent models of chaotic systems with multi-objective simulated annealing algorithms. Inf. Sci. 178, 952–970 (2008)

15. Slowik, A.: Fuzzy Control of Trade-off Between Exploration and Exploitation Properties of Evolutionary Algorithms. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS, vol. 6678, pp. 59–66. Springer, Heidelberg (2011)
16. Villar, J.R., Berzosa, A., de la Cal, E., Sedano, J., García-Tamargo, M.: Multi-objective learning of white box models with low quality data. Neurocomputing (in Press)
17. Villar, J.R., Otero, A., Otero, J., Sánchez, L.: Taximeter verification using imprecise data from gps. Eng. Appl. Artif. Intell. 22, 250–260 (2009)

A Simple Artificial Chemistry Model for Nash Equilibria Detection in Large Cournot Games

Rodica Ioana Lung and Lucian Sturzu-Năstase

Babeş-Bolyai University of Cluj Napoca, Romania
rodica.lung@econ.ubbcluj.ro, euristics@yahoo.com,

Abstract. A simple Artificial Chemistry model designed for computing Nash Equilibria for continuous games called Artificial Chemistry for Nash Equilibria (ACNE) is proposed. ACNE uses elementary reactions between strategy profiles of a noncooperative game with a generative relation for Nash Equilibria in order to solve noncooperative games. Experimental results indicating the potential of the new proposed method are performed on Cournot oligopolies for up to 1000 players and compared with those obtained by other methods.

Keywords: Artificial Chemistry, Nash Equilibrium, Cournot oligopoly.

1 Introduction

The problem of detecting Nash equilibria in large Cournot games represents a computational challenge when the number of players is increased and also because of well-known issues arising from the nature of the Nash equilibrium concept. From the first point of view, for example, a 1000 players game may be compared to a many-objective maximization optimization problem with 1000 objectives [4]. The natural connection between the two is that there are 1000 payoff functions/objective functions to be maximized. The main difference consists in the solution concept searched - the Nash equilibrium in the case of a game and the Pareto optimal solutions for the many-objective optimization problem. The Nash equilibrium is represented by a strategy profile such that no player can increase its payoff by unilaterally deviating. It is defined in the context of noncooperative games where no communication or cooperation between players is considered.

This paper presents a simple Artificial Chemistry model designed for detecting Nash equilibria in large Cournot games. The model simulates simple chemistry reactions where strategy profiles are considered as the molecules. The results are compared with those obtained by an Extremal Optimization algorithm for Nash Equilibria detection which are - to the best of our knowledge - the best results reported so far in the literature.

The paper is structured as follows: Section 2 presents a short introduction to the Artificial Chemistry field, Section 3 presents the formal definition of the Nash equilibrium and the generative relation used to detect it, the proposed method is presented in Section 4 and numerical experiments in Section 5.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 350–358, 2012.

2 Artificial Chemistries

Artificial Chemistries (AC) represent computational models inspired from the field of chemistry used for simulation or optimization. According to Dittrich et al. [3] there are two formal ways to define an artificial chemistry:

1. As a triplet (S, R, A) where S is the set of all possible molecules, R is the set of collision rules representing the interaction between molecules and A an algorithm describing the reaction vessel or domain and how the rules are applied to molecules inside the vessel;
2. An alternate definition as a tuple (S, I) where S is a set of particles and I is a description of the interactions among particles. This approach is preferable if interactions are taking place in the space of particles or molecules and it is the one used in this paper.

AC have been widely used in modeling, information processing and optimization. To the best of the authors knowledge it is the first time such an approach has been proposed to compute Nash equilibria in mathematical games.

3 Nash Equlibria - Definition and Generative Relation

Nash equilibrium [8] is the most popular solution concept in noncooperative game theory. A finite strategic game is defined by a set o players, a set of strategies available to each player and a set of payoff functions for each player and denoted 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.

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

A strategy profile $s \in S$ for the game Γ represents a Nash equilibrium [7,8] if no player has anything to gain by changing his own strategy while the others do not modify theirs.

Several methods to compute NE of a game have been developed. For a review on computing techniques for the NE see [7].

3.1 Nash Ascendancy Relation

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 which one is 'closer' to equilibrium. In [6] such a generative relation has been introduced and shown that solutions that are non-dominated/ascended with respect to this relation are exactly the Nash equilibria of the game.

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

$$k(s, s') = |(\{i \in \{1, ..., n\} | u_i(s'_i, s_{-i}) \geq u_i(s), s'_i \neq s_i\}|$$

to the pair (s, s') is introduced.

This set is composed by the players i that would benefit if - given the strategy profile s - would change their strategy from s_i to s'_i, i.e.

$$u_i(s'_i, s_{-i}) \geq u_i(s).$$

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

$$k(s, s') < k(s', s)$$

holds.

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

Two strategy profiles $s, s' \in S$ may have the following relation:

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

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 [6] 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 and can be considered as a generative relation for NEs.

4 Artificial Chemistry for Nash Equilibria

The following model inspired from elementary chemistry reactions is proposed. Consider the set of all possible situations of the game (strategy profiles S) as the molecule set M. The set of rules considers applying two elementary chemical reactions the *dissociation* and the *addition* reaction that are described in the following:

- The Addition reaction that merges two molecules: $A + B \rightarrow AB$.
- The dissociation of a molecule AB into fragments A and B: $AB \rightarrow A + B$;

Within the first step of ACNE four molecules (representing strategy profiles) are radomly sampled from the set M:

- A molecule R will play the elitist role of preserving the best solution found so far;
- The three other will play the role of A, B and AB, respectively.

4.1 The Addition Reaction

The addition reaction assumes that two molecules A and B merge into another one called AB. Within ACNE the following steps are performed to merge information from A and B:

1. AB is evaluated and the player j having the worst payoff is chosen;
2. The strategy of player j within AB is computed using a linear combination between corresponding strategies from A and B:

$$AB_j = rand() * A_j + rand() * B_j,$$

   where $rand()$ represents an uniformly generated number between 0 and 1;
3. If AB Nash ascends R then AB will replace R.

4.2 The Dissociation Reaction

The dissociation reaction is used to separate information in molecule AB in two others A and B. The following steps are followed:

1. AB is evaluated and player j having the worst payoff is chosen;
2. The strategy of player j within A is randomly generated;
3. R is also evaluated and player k having the worst payoff is chosen;
4. The strategy of player k within B is randomly generated;

Remarks

1. During the dissociation reaction the Nash ascendancy relation is not used, strategies corresponding to j and k are simply randomly generated without any other move.
2. It may be argued that during the dissociation reaction R is also involved. As R is randomly generated in the firs step of ACNE and after that it can be replaced only by AB if it is ascended by it, at one step R is either equal to AB or better than AB in Nash sense but also an 'old' AB and therefore we can consider that the information from R comes also from AB

Algorithm 1. Artificial Chemistry for Nash Equilibria

Randomly generate R, A, B, AB;
repeat
  Perform the Addition reaction;
  Perform the Dissociation reaction;
until a desired number of iterations is reached.
Return R.

ACNE is described in Algorithm 1. During an iteration ACNE performs the two types of reactions described above until a number of iterations (or a maximum number of payoffs) is reached. Another termination condition for ACNE that mimics a chemical phenomenon may be that the system composed of the four molecules reaches an equilibrium between the two reactions (there are no more improvements in R). An important remark is that the only parameter used by ACNE is the one involved in the termination condition (maximum number of iterations or maximum number of payoff functions evaluations).

5 Numerical Experiments

The construction of the two elementary reactions within ACNE is somewhat similar and actually inspired from an optimization method called Extremal Optimization (EO) [1]. An algorithm for Nash Equilibria detection based on EO called Nash Extremal Optimization has been proposed in [5] with very good numerical results for the Cournot oligopoly game [2]. In order to evaluate the performance of ACNE the Cournot oligopoly is used and results are compared with those from [5].

5.1 Cournot Oligopoly

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

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

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:

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

Table 1. Descriptive statistics of the results obtained using the two methods

Method	Avg	StdDev	Variance	Max	Min	Median
EO-10	2.7102	**0.01956**	**0.000382**	2.76586	2.702	2.70419
ACNE-10	**2.4382**	0.35803	0.12818	**2.70422**	**1.80287**	2.70384
EO-50	2.5459	**0.00465**	**2.16e-05**	2.55581	2.53882	2.54556
ACNE-50	**1.9284**	0.76406	0.58378	**2.55329**	**0.776151**	**2.41247**
EO-100	2.5316	**0.02944**	**0.000866**	2.60155	2.48389	2.52695
ACNE-100	**1.8085**	0.81461	0.6636	**2.52497**	**0.50949**	**2.147705**
EO-250	2.5304	**0.10911**	**0.011905**	2.69826	2.314	2.545455
ACNE-250	**1.7949**	0.83569	0.69838	**2.59191**	**0.565506**	**2.079055**
EO-500	3.3709	**0.48868**	**0.23881**	4.2192	2.38604	3.444135
ACNE-500	**2.7965**	0.5099	0.26	**3.44417**	**1.86609**	**2.95285**
EO-750	6.8425	0.45246	0.20472	7.66128	6.23826	6.76594
ACNE-750	**5.7895**	**0.30491**	**0.092969**	**6.35074**	**5.31033**	**5.80064**
EO-1000	12.394	0.44472	0.19777	13.1636	11.5602	12.4786
ACNE-1000	**10.58**	**0.2376**	**0.05648**	**10.8682**	**10.2429**	**10.54615**

If we consider

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

then the Cournot oligopoly has one Nash equilibria 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 represents a suitable and scalable benchmark for testing computational models for detecting Nash equilibria.

5.2 Parameter Settings

ACNE and EO are tested for 10, 50, 100, 250, 500, 750, and 1000 players. Because the number of payoff functions is equal to the number of players, this setting creates the equivalent of seven many-objective optimization problem which are known to be difficult to solve by evolutionary algorithms.

Both ACNE and EO were run 10 times and the average and standard deviation of the minimum distance to the NE for the 10 runs was computed. The stopping criterion is the maximum number of individual payoff functions evaluation of $2 \cdot 10^7$. None of the two methods use any other specific parameters.

5.3 Results

Table 1 presents descriptive statistics of the obtained results for the problem settings tested and associated boxplots are presented in Figs 1 and 2.

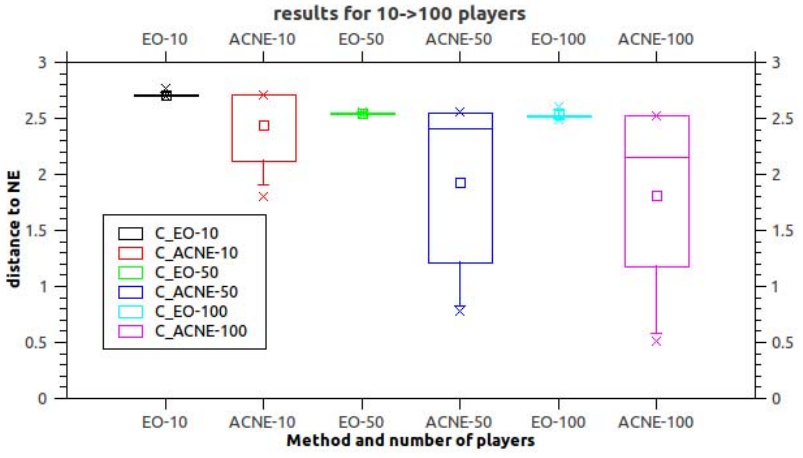


Fig. 1. Boxplots of the results obtained for 10, 50 and 100 players

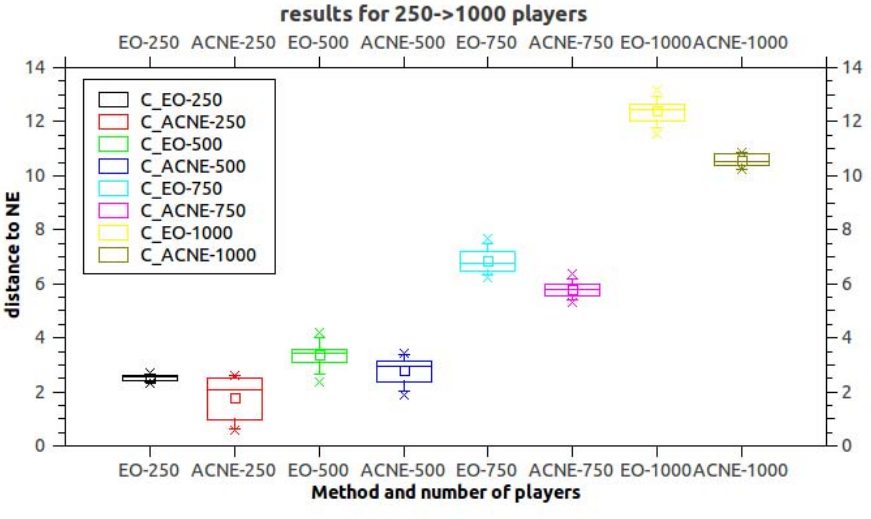


Fig. 2. Boxplots of the results obtained for 250, 500, 750 and 1000 players

Remarks

- Average distance to NE is smaller for ACNE in all seven instances;
- For smaller number of players (up to 100) the differences are not statistically significant;
- For 750 and 1000 players results obtained by ACNE are significantly better than those obtained using EO;

- Minimum and Maximum values obtained over the multiple runs are better for ACNE in all cases;
- Up to 250 players standard deviation and variance values indicate that the performance of EO is more stable;
- Interesting behavior:
  - When the number of players increases the results obtained with ACNE are more stable (standard deviation and variance values are decreasing);
  - For smaller number of players standard deviation for ACNE results are higher but this is due to the fact that there are very good values for the minimum values in the 10 runs considered (Fig. 1);

6 Conclusions and Further Work

A simple artificial chemistry model designed for solving noncooperative games is proposed. This model simulates two elementary reaction types: the addition reaction and the dissociation reaction considering the set of strategy profiles of the game as the set of molecules. Numerical experiments are performed on Cournot oligopolies with large number of players (up to 1000) and compared with an extremal optimization algorithm for Nash equilibria detection. Preliminary results indicate the potential of the method.

This is however only a very simple computational model inspired from elementary chemical reactions. In the future we plan to design a model for computing Nash equilibria in mixed form for matrix games.

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 and from the SECTORAL OPERATIONAL PROGRAMME HUMAN RESOURCES DEVELOPMENT, Contract POSDRU 6/1.5/S/3 Doctoral studies: through science towards society, Babes - Bolyai University, Cluj - Napoca, Romania.

References

1. Boettcher, S., Percus, A.G.: Optimization with Extremal Dynamics. Physical Review Letters 86, 5211–5214 (2001)
2. Daughety, A.F.: Cournot oligopoly: characterization and applications. In: Daughety, A.F. (ed.) Cambridge University Press, Cambridge (1988)
3. Dittrich, P., Ziegler, J., Banzhaf, W.: Artificial chemistries-a review. Artificial Life 7(3), 225–275 (2001)
4. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization. In: 3rd International Workshop on Genetic and Evolving Systems, GEFS 2008, pp. 47–52 (March 2008)
5. Lung, R.I., Mihoc, T.D., Dumitrescu, D.: Nash Extremal Optimization and Large Cournot Games. In: Pelta, D.A., Krasnogor, N., Dumitrescu, D., Chira, C., Lung, R. (eds.) NICSO 2011. SCI, vol. 387, pp. 195–203. Springer, Heidelberg (2011)

6. 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)
7. 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 (1996)
8. Nash, J.F.: Non-cooperative games. Annals of Mathematics 54, 286–295 (1951)

Dynamics of Networks Evolved for Cellular Automata Computation

Anca Gog and Camelia Chira

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

Abstract. Cellular Automata (CAs) represent useful and important tools in the study of complex systems and interactions. The problem of finding CA rules able to generate a desired global behavior is considered of great importance and highly challenging. Evolutionary computing offers promising models for addressing this inverse problem of global to local mapping. A related approach less investigated refers to finding robust network topologies that can be used in connection with a simple fixed rule in CA computation. The focus of this study is the evolution and dynamics of small-world networks for the density classification task in CAs. The best evolved networks are analyzed in terms of their tolerance to dynamic network changes. Results indicate a good performance and robustness of the obtained small-world networks for CA density problem.

Keywords: Cellular automata, density classification task, complex networks, evolutionary algorithms.

1 Introduction

Complex systems and their important principles of emergence, auto-organization and adaptability are intensively studied in fields such as physics, biology, computer science, sociology and economics. A complex system is characterized by the lack of central control while individual components are simple compared to the collective behavior [1]. The system behavior cannot be identified by considering each individual entity and combining them, but considering how the relationships between entities affect the behavior of the whole system.

Networks are a central model for the description of many complex phenomena. Typical examples of complex networks in nature and society include metabolic networks, the immune system, the brain, the human social networks, communication and transport networks, the Internet and the World Wide Web. The study of real-world networks revealed features as degree distribution, average distance between vertices, network transitivity and community structure [2,3,4,5].

The analysis of networks focuses on their topological structure which highly influences network processes [6,7]. Watts [8] observed that most real-world networks have structural properties that make them different from regular lattices

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 359–368, 2012.

and random graphs. One such property is the "small world effect" i.e. the average distance between vertices in a network is short, usually scaling logarithmically with the number of vertices in the network. Besides the presence of short paths, *small-world networks* are characterized by a large clustering coefficient i.e. the probability that two vertices that are both neighbors to the same third vertex are also neighbors of each other (also called network transitivity). This means that small-world networks have more local structure compared to random graphs. Watts and Strogatz [9,8] studied the computational properties of small-world networks by examining Cellular Automata (CAs) computation on this type of networks. In [10,7,6], small-world type network topologies are evolved starting from an initial population of regular and random structures. In these studies, it is shown that the evolved topologies have better performance for the CA majority and synchronization problems than regular lattice structures.

In this paper, we investigate the evolution and dynamics of small-world networks for CA computation. The density classification task (also called the majority problem) is addressed using network topologies evolved based on a simple standard genetic algorithm. The best-fitted networks are further analyzed in terms of their robustness to dynamics by evaluating their performance as the topology is perturbed for the density classification task. Computational experiments and results indicate that the obtained networks have a competitive performance for CA density classification even when simply the majority rule is applied. Furthermore, perturbations on the network topology do not cause the failure of the system and the performance is only slightly affected by network dynamics.

The rest of the paper is structured as follows: sections 2 and 3 describe CAs and the density classification problem as well as related work in the area of network based CA computation, section 4 presents the computational approach to evolve network topologies, discusses the numerical experiments performed and presents the analysis of network dynamics for the density task, and, finally, section 5 contains the conclusions and directions for future research.

2 Cellular Automata Computation: Density Classification Task

CAs are decentralized structures of simple and locally interacting elements (cells) that evolve following a set of rules [11]. Programming CAs is not an easy task, particularly when the desired computation requires global coordination. CAs provide an idealized environment for studying how (simulated) evolution can develop systems characterized by "emergent computation" where a global, coordinated behavior results from the local interaction of simple components [12].

The one-dimensional binary-state CA capable of performing computational tasks has been extensively studied in the literature [13,14,15,16,1]. Normally, a one-dimensional lattice of N two-state cells is used for representing the CA. The state of each cell changes according to a function depending on the current states in the neighborhood. The neighborhood of a cell is given by the cell itself and

its r neighbors on both sides of the cell, where r represents the radius of the CA. The initial configuration of cell states (0s and 1s) for the lattice evolves in discrete time steps updating cells simultaneously according to the CA rule.

One of the most widely studied CA problems is the density classification task (DCT). The aim of DCT is to find a binary one-dimensional CA able to classify the density of 1s (denoted by ρ_0) in the initial configuration. If $\rho_0 > 0.5$ (1 is dominant in the initial configuration) then the CA must reach a fixed-point configuration of 1s otherwise it must reach a fixed-point configuration of 0s within a certain number of time steps. Most studies consider the case $N = 149$ (which means that the majority is always defined) and neighborhood size of 7 (the radius of CA is $r = 3$). The CA lattice starts with a given binary string called the initial configuration (IC). After a maximum number of iterations (usually set as twice the size of CA), the CA will reach a certain configuration. If this is formed of homogeneous states of all 1s or 0s, it means that the IC has been classified as density class 1, respectively 0. Otherwise, the CA makes by definition a mis-classification [17]. It has been shown that there is no rule that can correctly classify all possible ICs [18].

The performance of a rule measures the classification accuracy of a CA based on the fraction of correct classifications over 10^4 ICs selected from an unbiased distribution (ρ_0 is centered around 0.5).

DCT is a challenging problem extensively studied due to its simple description and potential to generate a variety of complex behaviors. Most studies focused on developing algorithms able to find high performant rules for 1D CAs with fixed neighborhood size. The performance of the best rules obtained ranges from 0.76 (obtained with genetic algorithms [19,12,17,20], 0.82 (obtained with genetic programming [21]) to 0.86 (detected by coevolutionary learning [13,22] and multiobjective evolutionary model [23]) and 0.889 (found by a two-tier evolutionary framework [24]).

3 Network Based CAs

A few studies [9,8,10,7,6] consider an extension of the CA concept in which the cells can be connected in any way while the rule is the same for all cells. In this approach, the topological structure of CAs refers to general graphs.

Watts [8] studied the small-world graph version of the DCT: the rule is fixed and the performance of different small-world networks for DCT is evaluated. A small-world graph is constructed starting from a regular ring of nodes in which each node has k neighbors. A random rewiring procedure is then applied as follows [9,8]: a vertex and the edge connecting it to a neighbor is chosen, and the edge is reconnected with probability p to a vertex uniformly chosen at random from the entire ring. This process is repeated by moving clockwise around the ring until each vertex is considered once (in connection with nearest neighbors, second-nearest neighbors, etc. - depending on the value of k). This way, a number of shortcuts (i.e. edges that link nodes which would be more than

two edges apart if they were not directly connected) are produced. Watts and Strogatz [9] observe that for intermediate values of p the graph constructed in this way is a small-world network (with small characteristic path length and high clustering coefficient).

The rule used for small-world network DCT is simple: at each time step, each node takes the state of the majority of its neighbor nodes in the graph (if the number of state 1s equals the number of state 0s in the neighbors list then the node is randomly assigned a state with equal probability between 0 and 1). Small-world networks proved to have a performance of around 0.8 for the DCT with this fixed majority rule for 149 cells CA.

Tomassini et al [10,7,6] investigated network based CAs for the density and synchronization problems. They use spatially structured evolutionary algorithms to find the best performant network topology for DCT when the rule is fixed to the majority rule described above. An individual represents a network structure and the fitness is computed based on the fraction of ICs (out of 100 ICs generated anew for each individual) correctly classified by the majority rule based on the neighborhood given by the network. The initial population is generated in two ways: starting from regular rings with node degree $k = 4$ (slightly perturbed by adding a link for each node with a low probability) and random graphs. The best evolved network starting from initial regular rings has a performance of 0.823 (for 149 cells) while the result for random graphs as initial population is similar (performance of 0.821 of the best network). These results are triggered by the same majority rule used by Watts [9,8]. An important contribution of [10,7,6] is defining the computational model for evolving small-world networks for DCT. Equal or better performance can be obtained when a constraint regarding the proportion of shortcuts is added via the fitness function (so as to favor networks with low values for the fraction of edges that are shortcuts) [6]. The robustness of the evolved topologies is tested in two scenarios as follows: (i) *probabilistics faults* [6] which allow the rule of each cell to yield an incorrect output state with a certain probability (the structure of the network is not affected in any way in this situation), and (ii) *permanent link failures* [10] defined as the definitive disappearance of an edge between two nodes of the graph. Results obtained indicate that irregular networked automata show an outstanding robustness and are more tolerant to faults compared to lattice CAs for the density and synchronization tasks [7].

4 Evolution and Dynamics of Network Topologies for DCT

We present the development of a simple evolutionary algorithm to produce network topologies (of different sizes) which are tested in the DCT based on the majority rule. The robustness of the best-fitted networks is assessed by evaluating their performance as the topology is perturbed for the DCT.

4.1 Evolving Network Topologies for DCT

A standard evolutionary algorithm [25,26] is engaged to evolve small-world topologies for cellular automata with the fixed majority rule described in the previous section. An individual represents a network and is encoded as an array of integers representing nodes and a list of links for each node. The evolution process starts with a regular lattice ring where each cell has 4 neighbors. The population has 100 individuals and the number of generations is 100. The elitism probability is 10%. For the other individuals a standard proportional selection is applied and each selected individual undergoes mutation. Applied for each node of the chromosome with probability 0.5, mutation randomly selects a link of the current node, removes it and reconnects the node with another randomly selected node. These actions take place with the restrictions of not having duplicated links and not having nodes with no links. The fitness function is computed as the fraction of correctly classified 100 randomly generated initial configurations.

This algorithm has been applied for evolving network topologies for cellular automata with 49 cells, 99 cells and 149 cells. The performance of the obtained topologies based on the fixed majority rule is really good: around 0.85 for 49 cells, 0.83 for 99 cells and 0.82 for 149 cells. This result (for CA size 149) is similar to the one reported in [10,6] but it has been obtained using a standard genetic algorithm as opposed to spatially structured evolutionary algorithms.

For all three obtained networks of size 49, 99 and 149 the average node degree is 6, which resembles the neighborhood of 7 usually used for regular lattice CAs. Figure 1 presents the distribution of node degrees in the obtained networks. It can be noticed that most of the node degrees are close to the average.

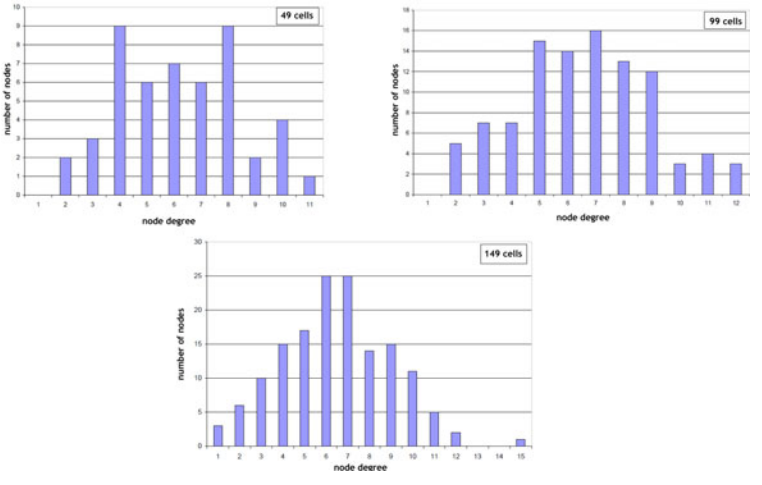


Fig. 1. Node degree distribution for network topologies with 49 cells, 99 cells and 149 cells

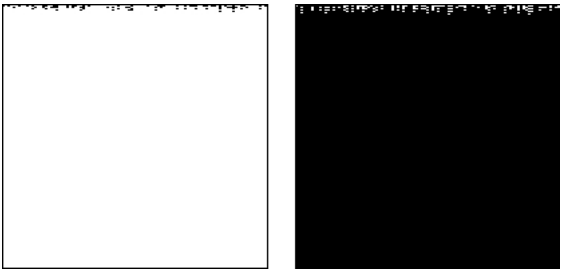


Fig. 2. Evolution of 99 cells CA with best network topology produced by the evolutionary algorithm and the fixed majority rule

Figure 2 presents the evolution of a 99 cells CA based on network topology with the majority rule. In the left side of the figure, the initial configuration has a majority of white cells and the fixed majority rule based on the neighborhood given by the evolved network correctly classifies the input within a few time steps. In the right side, an initial configuration with 1s majority cells is correctly classified.

4.2 Network Dynamics Analysis

In order to evaluate the robustness of the evolved network topologies (described in the previous section), they have been subject to dynamic changes understood as random removal or addition of network links. It should be noticed that we also consider the possibility of adding an edge between two nodes as opposed to permanent link failures tested in [10] where only edge removals are considered.

The number of dynamic changes equals the number of edges in the network. At each step, a randomly selected link is either removed or added to the network with probability p and the resulting network is again evaluated with respect to the performance. Figure 3 depicts the performance of the considered networks subject to dynamic changes with probability $p = 0.5$. All three networks (of size 49, 99 and 149) are still able to trigger good performances on the DCT i.e. above 0.81. The highest fluctuation in performance is observed in the 49 cells network. This might be due to the fact that the algorithm has been able to find a good solution for this small network in its 100 generations, and a certain degree of destabilization is induced when the network is subject to further dynamic changes.

We further analyze the performance of the best evolved network (for each size considered) when the probability p of a dynamic change ranges from 0.1 to 1 (see Figures 4-6). For all networks, the performance for DCT remains good even when changes are induced with high probability. This is an indication of the robustness of the evolved network topologies and their capability to obtain better results for the DCT (which can be potentially further improved in connection with more

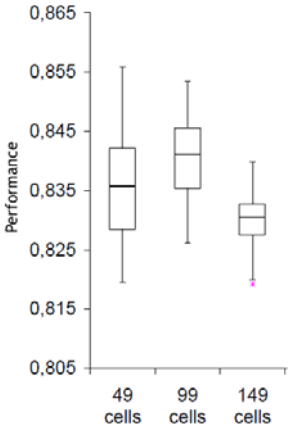


Fig. 3. The performance of the three networks (of size 49, 99 and 149) that are subject to dynamic changes with probability $p = 0.5$

sophisticated rules compared to the fixed majority rule). For higher size networks (particularly for 149 cells), a performance increase is registered as dynamics are induced in the network structure. This result can be potentially explained by the fact that the networks obtained after 100 generations of the evolutionary algorithm can be further improved and some dynamic changes randomly applied might lead to better network topologies for DCT.

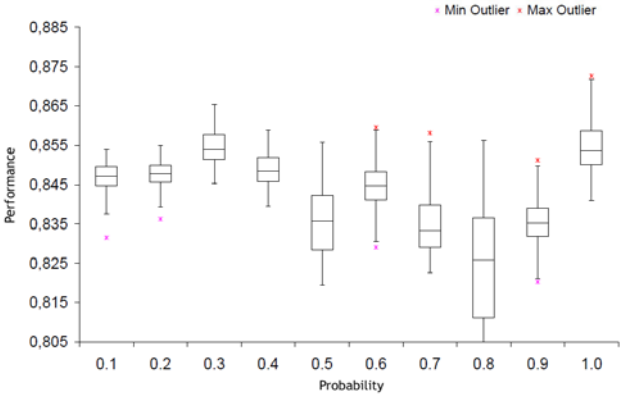


Fig. 4. The performance of the 49 cells network subject to dynamic changes with different probabilities

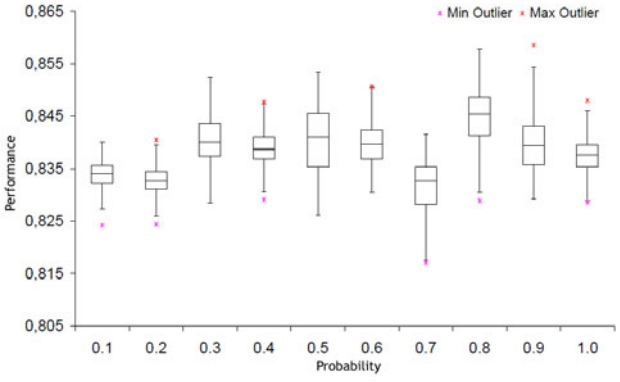


Fig. 5. The performance of the 99 cells network subject to dynamic changes with different probabilities

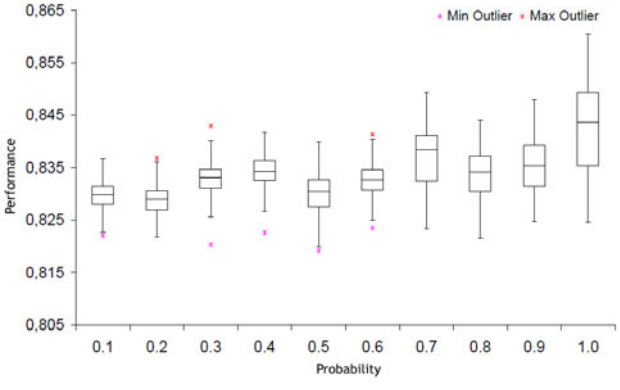


Fig. 6. The performance of the 149 cells network subject to dynamic changes with different probabilities

5 Conclusions

A standard evolutionary algorithm has been applied for evolving network topologies for cellular automata. The performance of the evolved networks has been tested against dynamic changes understood as random removal or addition of links between nodes with different probabilities. The overall performance of the perturbed networks does not show any significant fluctuations emphasizing the robustness of the evolved networks.

This study will be further extended by analyzing the structural properties of the evolved networks and by applying dynamic changes with separate probabilities for removal and addition of links. Comparisons with other types of networks

and with regular lattices that are subject to dynamic changes will also be performed for the majority task as well as for other CA tasks.

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 CNCS Romania.

References

1. Chira, C., Gog, A., Lung, R.I., Iclanzan, D.: Complex Systems and Cellular Automata Models in the Study of Complexity. Studia Informatica, vol. LV(4), pp. 33–49 (2010)
2. Barabasi, A.-L.: Linked: The New Science of Networks. Perseus, New York (2002)
3. Watts, D.J.: Six degrees: The Science of a Connected Age. Gardner's Books, New York (2003)
4. Newman, M.E.J., Girvan, M.: Finding and Evaluating Community Structure in Networks. Physical Review E 69, 026113-1 (2004)
5. Girvan, M., Newman, M.E.J.: Community Structure in Social and Biological Networks. Proceedings of the National Academy of Sciences of the USA 99, 7821–7826 (2002)
6. Tomassini, M., Giacobini, M., Darabos, C.: Evolution and dynamics of small-world cellular automata. Complex Systems 15, 261–284 (2005)
7. Darabos, C., Tomassini, M., Di Cunto, F., Provero, P., Moore, J.H., Giacobini, M.: Toward robust network based complex systems: from evolutionary cellular automata to biological models. Intelligenza Artificiale 5(1), 37–47 (2011)
8. Watts, D.J.: Small Worlds: The Dynamics of Networks Between Order and Randomness. Princeton University Press, Princeton (1999)
9. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'smallworld' networks. Nature 393, 440–442 (1998)
10. Darabos, C., Giacobini, M., Tomassini, M.: Performance and Robustness of Cellular Automata Computation on Irregular Networks. Advances in Complex Systems 10, 85–110 (2007)
11. Wolfram, S.: Theory and Applications of Cellular Automata. Advanced series on complex systems, 9128. World Scientific Publishing (1986)
12. Mitchell, M., Crutchfield, J.P., Das, R.: Evolving cellular automata with genetic algorithms: A review of recent work. In: Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA 1996). Russian Academy of Sciences (1996)
13. Juille, H., Pollack, J.B.: Coevolving the 'ideal' trainer: Application to the discovery of cellular automata rules. In: Proceedings of the Third Annual Conference on Genetic Programming (1998)
14. Tomassini, M., Venzi, M.: Evolution of Asynchronous Cellular Automata for the Density Task. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 934–943. Springer, Heidelberg (2002)
15. Mitchell, M., Thomure, M.D., Williams, N.L.: The role of space in the Success of Coevolutionary Learning. In: Proceedings of ALIFE X - The Tenth International Conference on the Simulation and Synthesis of Living Systems (2006)

16. Oliveira, G.M.B., Martins, L.G.A., de Carvalho, L.B., Fynn, E.: Some investigations about synchronization and density classification tasks in one-dimensional and two-dimensional cellular automata rule spaces. Electron. Notes Theor. Comput. 252, 121–142 (2009)
17. Pagie, L., Mitchell, M.: A comparison of evolutionary and coevolutionary search. Int. J. Comput. Intell. Appl. 2(1), 53–69 (2002)
18. Land, M., Belew, R.K.: No perfect two-state cellular automata for density classification exists. Physical Review Letters 74(25), 5148–5150 (1995)
19. Das, R., Mitchell, M., Crutchfield, J.P.: A Genetic Algorithm Discovers Particle-Based Computation in Cellular Automata. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 344–353. Springer, Heidelberg (1994)
20. Gog, A., Chira, C.: Cellular Automata Rule Detection Using Circular Asynchronous Evolutionary Search. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 261–268. Springer, Heidelberg (2009)
21. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press (1992)
22. Juille, H., Pollack, J.B.: Coevolutionary learning and the design of complex systems. Advances in Complex Systems 2(4), 371–394 (2000)
23. de Oliveira, P.P.B., Bortot, J.C., Oliveira, G.: The best currently known class of dynamically equivalent cellular automata rules for density classification. Neurocomputing 70(1-3), 35–43 (2006)
24. Wolz, D., de Oliveira, P.P.B.: Very effective evolutionary techniques for searching cellular automata rule spaces. Journal of Cellular Automata 3, 289–312 (2008)
25. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Kluwer Academic Publishers, Boston (1989)
26. Dumitrescu, D., Lazzerini, B., Jain, L.C., Dumitrescu, A.: Evolutionary Computation. CRC Press, Boca Raton (2000)

From Likelihood Uncertainty to Fuzziness: A Possibility-Based Approach for Building Clinical DSSs

Marco Pota, Massimo Esposito, and Giuseppe De Pietro

Institute for High Performance Computing and Networking
ICAR-CNR
via P. Castellino 111, Naples 80131, Italy
{marco.pota,massimo.esposito,giuseppe.depietro}@na.icar.cnr.it

Abstract. For data classification, in fields like medicine, where vague concepts have to be considered, and where, at the same time, intelligible rules are required, research agrees on utility of fuzzy logic. In this ambit, if statistical information about the problem is known, or can be extracted from data, it can be used to define fuzzy sets and rules. Statistical knowledge can be acquired in terms of probability distributions or likelihood functions. Here, an approach is proposed for the transformation of likelihood functions into fuzzy sets, which considers possibility measure, and different methods arising from this approach are presented. By using real data, a comparison among different methods is performed, based on the analysis of transformation properties and resulting fuzzy sets characteristics. Finally, the best method to be used in the context of clinical decision support systems (DSSs) is chosen.

Keywords: DSS, Fuzzy sets, fuzzy logic, likelihood, probability, possibility, transformation.

1 Introduction

Decision-making, especially in fields where vague concepts have to be handled, represents a very challenging issue. The main purpose of recent research efforts, in medicine, economy and automated processes control, is the classification of data items in a finite number of conclusions, in other words, the determination of the membership of an object to a specific class. An instrument skilled to help user's decision-making is called Decision Support System (DSS). Using data acquired from actual cases, whose membership to a class is known, a certain amount of knowledge can be acquired, about the relationships between data items and respective class membership. If this knowledge is properly modeled, new incoming data items can be classified. In knowledge-based DSSs, data are processed and modeled by exploiting a rule representation formalism, while in data-driven DSSs, they are used as training set for statistical and machine learning models. A number of particular approaches exists, consisting in various modules of both knowledge-based and data-driven DSSs, and hybrid combination of them.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 369–380, 2012.

In clinical decision processes, widespread agreement is gained by researchers on the utility of DSSs based on fuzzy logic [1]. The main strength points of fuzzy logic are related to the ability of managing uncertainty and vagueness, which are typical in this field, and to the transparency and comprehensibility of its knowledge base. These properties are considered very attractive, because they allow a medical expert to study and interpret the resulting rules and linguistic variables and terms, and to eventually improve or adapt them according to previous knowledge or further experimental data.

The construction of DSSs based on fuzzy logic requires the definition of fuzzy sets.

Rough definitions could be known a priori, or specified by decision tree algorithms [2]. Improved membership functions could be obtained by optimization algorithms, in order to maximize the "goodness" (e.g., the classification rate) of the DSSs [2]. However, some problems can arise with the optimization approach, related to i) the prior definition of the sets' shape, ii) the risk of overfitting, and/or iii) the disagreement between labels and ranges of optimized sets.

A different approach consists in the generation of fuzzy sets definitions by using statistical information. This is a solution to a greater set of problems, with respect to the approach mentioned above, because statistical information, like probability distributions or likelihood functions, can be extracted from data by means of statistical analysis and clustering techniques [3], but also, especially in medical fields, can be promptly acquired from previous knowledge. Moreover, this represents an appealing solution to the above mentioned tasks: *i)* in this case the assumption about the shape of the involved functions is not obligatory (it is also possible to calculate any single point); *ii)* the risk of overfitting can be avoided by using smooth functions; iii) the ranges of fuzzy sets are calculated according to statistic functions, which correspond to common knowledge about the phenomenon, hence they are in agreement with respective linguistic labels. Finally, since the final users, like physicians in medical settings, are usually skilled to think and work according to a statistical interpretation of knowledge, this approach can significantly reduce the existing lack of familiarity shown by physicians in thinking in a fuzzy fashion.

In the ambit of fuzzy set definition starting from statistical information, two problems have typically to be solved.

The first challenge concerns the transformation of a probability distribution into a fuzzy set. Starting from the knowledge of a probability distribution, the construction of a fuzzy set has been widely studied, and inherent literature will be detailed in the following. Given a category 'F_i' and a set X of x values for a random variable, the uncertain knowledge about the random variable, made explicit by the probability distribution of x (which is the function $\mathcal{D}_i(x) = p(x|F_i)$ defined on X), is usually translated into the vague knowledge of the limits of the fuzzy set F_i. The core of the transformation consists in obtaining a possibility distribution (the function $\pi(x|F_i)$ defined on X), where the possibility of occurrence of an event is a number assessing someone's knowledge about this possibility [4]. Once this step is made, the membership function of the fuzzy set coincides with the possibility distribution. Therefore, the resulting fuzzy set is obtained by using "random set view" [5] construction, and can be described as follows: the membership grade $\mu_{F_i}(x)$ is the degree of truth associated to the statement:

"*If the fuzzy set is F_i, then the variable value is x*".

In this framework, possibility is considered to be an upper limit for the probability [6].

Very less attention was paid to the second problem to deal with, which regards the transformation of a likelihood function into a fuzzy set. This has also been treated in literature, and a direct method goes with a definition of fuzzy sets. Given n categories 'F_i', $i = 1, \dots, n$ and a set X of x values for a random variable, the likelihood function $\mathcal{L}_i(x) = p(F_i|x)$, defined on X, coincides with the membership function of the fuzzy set F_i [6]. While making this assumption, Dubois & Prade also admit that it is related to the fuzzy set interpretation in terms of conditional uncertainty measure, and that other measures could be used, in particular a possibility measure $\Pi(F_i|x)$. Therefore, the problem can be translated into the evaluation of such possibility measures. The associated membership function, which in this case coincides with the set of possibilities, is intended according to the so-called "likelihood view" [5], which is described as follows: the membership grade $\mu_{F_i}(x)$ is the degree of truth associated to the statement:

"*If the variable value is x, then the fuzzy set is F_i*".

This interpretation of a fuzzy set is typically used into fuzzy DSSs, where new data are given in the form of numeric values and have to be translated into fuzzy items.

The two probability-oriented views (upper probability and likelihood) of fuzzy sets and possibility distributions are not antagonistic [6].

Here, an approach is proposed which allows to obtain fuzzy sets, starting from likelihood-based knowledge. The fuzzy set interpretation in terms of possibility measure is obtained, as the problem of transforming a likelihood function into a set of possibilities is considered. In particular, a family of likelihood functions $\mathcal{L}_i(x)$, such that $\forall x, \sum_i \mathcal{L}_i(x) = 1$, is processed: for each x value, the set of $\mathcal{L}_i(x) = p(F_i|x)$, $i = 1, \dots, n$ is regarded as a (discrete) probability distribution. This is transformed into the associated possibility distribution, $\Pi(F_i|x)$, defined for the different classes F_i.

The application of the proposed approach is presented, and executed on a dataset obtained by a clinical case study, regarding the diagnosis of Multiple Sclerosis (MS).

The results are compared with the ones obtained by means of the traditional approach, which settles membership functions equal to likelihood.

Furthermore, since different methods can be applied to the probability-possibility transformation implied by the proposed approach, each of them is examined in order to check the satisfaction of principles of probability-possibility transformations, in the case of discrete probability distributions. Moreover, a comparative study of all the results is presented and discussed, by characterizing resulting fuzzy sets.

The rest of the paper is organized as follows. Section 2 summarizes the properties and the most representative examples of probability-possibility transformations, and the main characteristics of fuzzy sets. In section 3, the proposed approach is presented, together with all its variants. In section 4, different variants are compared, and evaluated with respect to the existing approach. Finally, section 5 concludes the work.

2 Related Concepts

2.1 Properties of Probability-Possibility Transformation

The transformations between functions describing probability and possibility have been widely studied, starting from the birth of possibility theory developed by Zadeh [7].

Several works start from a probability distribution $\mathcal{D}(y) = p(y|x)$, defined on Y, where y varies in a subset Y of the universe of discourse U and x is a specified value of the condition. In this section, $p(y)$ is written instead of $p(y|x)$ since the condition is fixed when a probability distribution is considered.

Most of the methods essentially transform $\mathcal{D}(y)$ into a possibility distribution $\pi: Y \rightarrow [0,1]$, which allows the measurement of the possibility $\Pi(A)$ of any finite subset A of Y. Distinct solutions have been found, each of them based on a choice of some of the following assumptions:

- **Normalization.** The possibility distribution has to be normalized [8] to ensure $\Pi(Y) = 1$. Therefore:

$$\exists y \in Y | \pi(y) = 1 \ . \tag{1}$$

- **Consistency.** Probability-possibility consistency has to be held [7]. The possibility measures can encode upper probabilities, therefore possibility degrees cannot be less than degrees of probability:

$$\forall A, P(A) \leq \Pi(A) \ . \tag{2}$$

  Consistency furnishes each possibility degree with a lower bound.
- **Specificity.** In order to preserve as much information as possible, when the transformation is performed, specificity has to be maximized. In other words, cardinality

$$card = \sum_i \pi(y_i) \tag{3}$$

  has to be minimized [8]. Maximal specificity is reached when cardinality is minimized and consistency is respected at the same time.
- **Uncertainty Invariance.** Another approach to the same concept of preserving information [9, 10] implies that invariance has to be held between uncertainties encoded by probability and possibility distributions, in other words the entropy of probability distribution should equal the energy of the possibility distribution:

$$H(p) = E(\pi) \ . \tag{4}$$

  This assumption is debatable, because it is based on the prerequisite that possibilistic and probabilistic information measures are commensurate.
- **Equidistribution.** Plausibility, as defined in evidence theory, can be approximated by probability [11]:

$$Pl(A) \cong P(A) \ . \tag{5}$$

- **Order Preservation.** Since the more one event is probable, the more possible it should be as well, preference has to be preserved [8]:

$$\pi(y_i) > \pi(y_j) \leftrightarrow p(y_i) > p(y_j) \ . \tag{6}$$

  One could refer to weak order-preservation if only p order implies π order.
- **Scaling.** A scaling assumption [4, 12, 13] forces each possibility value $\pi(y_i)$ to be a function of only the probability $p(y_i)$ of the same event:

$$\pi(y) = f(p(y)) \ . \tag{7}$$

  The function f can be ratio-scale, Log-interval scale, etc. However, such an assumption can lead to not consistent transformations [14].

2.2 Probability-Possibility Transformations

The most representative transformations can be summarized by the respective formulas, given below. Discrete probability distributions are considered here, with variable values indexed in such a way that

$$p(y_1) \geq p(y_2) \geq \cdots \geq p(y_n) \ . \tag{8}$$

Transformation 1

$$\pi(y_i) = \sum_{k=1}^{n} \min[p(y_i), p(y_k)] \tag{9}$$

This was firstly proposed by Dubois and Prade [4], is invertible and based on normalization, consistency and order preservation. It does not meet maximum specificity. Equivalent results were achieved by Yager and Kreinovich [9], who found a transformation based on uncertainty invariance, and by Yamada [15], who based his results on equidistribution.

Transformation 2

$$\pi(y_i) = \sum_{k=i}^{n} p(y_k) \tag{10}$$

This was developed by Dubois et al. [8], and is based on normalization, consistency, order preservation, and additionally maximum specificity.

Transformation 3

$$\pi(y_i) = \left(\frac{p(y_i)}{p(y_1)}\right)^{\alpha} \tag{11}$$

This was proposed by Klir [12] and is based on uncertainty invariance and scaling assumption. The exponent depends on the distribution and can be computed by imposing condition (4).

Transformation 4. Some works start from a probability distribution to reach a set of possibilities by statistical methods. Some of them use particular assumptions, while a method was given in our previous work [3], where a generalization of these existing methods is formulated, and theoretical support is proposed so as to justify these kinds of transformation.

If properly applied to unimodal, symmetric and non-constant probability distributions, this method gives the same result of method 2. Therefore, the transformation satisfies normalization, consistency, order preservation, and maximal specificity. Moreover, the algorithm forces the resulting set of possibilities to be unimodal, and a bounded support can be assigned. For other properties, see [16].

Anyway, this method can be only applied if an order relation is defined on the variable values.

2.3 Fuzzy Sets Characterization

The main characteristics which describe a fuzzy set can be summarized as follows.

- **Shape.** The fuzzy set can have a particular shape (triangular, trapezoidal, bell-shaped, Gaussian, ...); assumptions are usually made in order to calculate the membership functions with a finite number of degrees of freedom.
- **Bounded support.** The support of the fuzzy set can be limited or not; for calculation purposes, it is convenient to introduce cutoffs if an unlimited support is concerned.
- **Normality.** If there exists a variable value such that the membership function is 1, then the fuzzy set is said to be "normal". Otherwise, it is called "sub-normal". When the membership function equals a normalized possibility distribution, the corresponding fuzzy set is normal as well.
- **Orthogonality.** If more than one fuzzy set is concerned (which is the case, e.g., of different terms associated to the same linguistic variable, or of different possible classes associated to the same feature), they are called "orthogonal" if the sum of the membership grades of different fuzzy sets is 1, for each of the possible variable values.
- **Continuity & Derivability.** The membership function, and its first derivate, can be continuous or not; the application of particular computational procedures can require the continuity or the derivability. Discontinuous functions have to be avoided, since they are scarcely intelligible.
- **Interpretability.** The set of possibilities $\Pi(y|x)$, for different values of the first variable y, defines a fuzzy set in terms of the "random set view". On the other hand, the set of possibilities for different values of the condition x, defines a fuzzy set in terms of the "likelihood view". If the fuzzy set does not correspond to a set of possibilities but to a likelihood function, it is still interpretable with the likelihood view, and information encoded regards conditional uncertainty instead of vagueness.

3 The Proposed Approach

The focal point of this work regards the transformation from a likelihood function to a fuzzy set.

The first method, explicated by Dubois & Prade [6], is based on the assumption that

$$\mu_{F_i}^0(x) = \mathcal{L}_i(x) = p(F_i|x) \ . \tag{12}$$

Resulting membership grades are denoted here by the apex 0, and this approach is mentioned as "method 0".

Nevertheless, a different interpretability can be assigned to the fuzzy set. In particular, the construction of the membership function should take into account that the interpretation of the fuzzy sets, in the ambit of DSSs, is often needed in terms of vague definition and likelihood view. Therefore, an approach is presented here which allows to obtain the fuzzy set whose interpretation is the one required for DSSs purposes, in other words, for all of x values, the membership function is constructed in such a way that:

$$\mu_{F_i}(x) = \pi(F_i|x) \ . \tag{13}$$

In order to do this, a previous consideration has to be pointed out. Likelihood functions are never alone, a family of a number n of them ever exists instead. If only one likelihood function is explicitly defined, another one can be defined. In general, if a number $n-1$ of classes, and respective likelihood functions, are explicitly defined, a further class can be defined which comprises all the occurrences not comprised in the defined classes, and its likelihood function can be defined by imposing.

$$\mathcal{L}_n(x) = 1 - \sum_{i=1}^{n-1} \mathcal{L}_i(x) \ . \tag{14}$$

Of course, given a point x, the sum of likelihood functions in that point must equal 1.

The proposed approach is based on the fact that a family of likelihood functions $\mathcal{L}_i(x)$, $i = 1, \dots, n$ individuates for each x value a discrete probability distribution, i.e. the set of probabilities $p(F_i|x)$ for $i = 1, \dots, n$. Therefore, in correspondence of each x value, the discrete probability distribution can be transformed into a possibility distribution $\pi(F_i|x)$. Therefore, the membership grades of different fuzzy sets at the point x can be obtained according to equation (13). The membership function of fuzzy set F_i is thus obtained by considering the set of $\mu_{F_i}(x)$ membership grades for all of x values.

The transformation of the set of values of likelihood functions in the point x, i.e. a probability distribution, into a possibility distribution, can be made by means of different methods, well known in literature. Among those summarized in section 2.2, only transformations 1, 2 and 3 can be applied, since the other can be applied only for ordered variables, while the set of values of the variable having the probability distributions considered here is the set of classes F_i, which in general is not ordered.

Therefore, the complete method for transforming a likelihood function into a fuzzy set is made of a composition of the approach described above with one of the existing probability-possibility transformations. The following equations are given, which describe the approach, composed by probability-possibility transformations.

$$\mu_{F_i}^1(x) = \sum_{k=1}^{n} \min[p(F_i|x), p(\Phi_k|x)] \tag{15}$$

$$\mu_{F_i}^2(x) = \sum_{k=k_i}^{n} p(\Phi_k|x) \tag{16}$$

$$\mu_{F_i}^3 = \left(\frac{p(F_i|x)}{p(\Phi_1|x)}\right)^{\alpha} \tag{17}$$

where, for each x, Φ_k are the classes F, sorted in such a way that

$$p(\Phi_1|x) \geq p(\Phi_2|x) \geq \cdots \geq p(\Phi_n|x) \ , \tag{18}$$

k_i is the index such that $\Phi_{k_i} \equiv F_i$, and α exponent depends on the distribution and can be computed by imposing condition (4). In order to distinguish the obtained membership functions among them and from μ^0 obtained by Dubois and Prade assumption of equation (12), apexes 1, 2 and 3 are associated with μ symbol, which refer to the different probability-possibility transformations, described in section 2.2. Applications of transformations to this approach are respectively named “method 1”, “method 2” and ”method 3”.

4 Comparison of Likelihood – Fuzzy Set Transformations

In order to describe and compare different methods for transforming likelihood function into fuzzy set, given in equations (12), (15), (16) and (17), a real clinical situation was considered, regarding the classification of Multiple Sclerosis lesions. The experimental dataset, coming from Magnetic Resonance images, is made of four features describing the brain tissues in the form of the following variables: white matter fraction (WM) representing the percentage of white matter surrounding a tissue, a shape factor (SF), a distance factor (DF) modeling the minimum color contrast to determine a tissue, and the volumetric dimension (VN) expressed in number of voxels (i.e. volumetric pixels). Furthermore, dataset is labeled with respect to two different classes, namely normal brain tissues (NBTs) and clusters of potentially abnormal white matter, called White Matter Potential Lesions (WMPLs).

Different datasets with more than two classes could be used, and the same conclusions about the proposed approach would be obtained.

The values of each variable of the experimental dataset were separated depending on the different classes, which in this case correspond to *NBT* and *WMPL*.

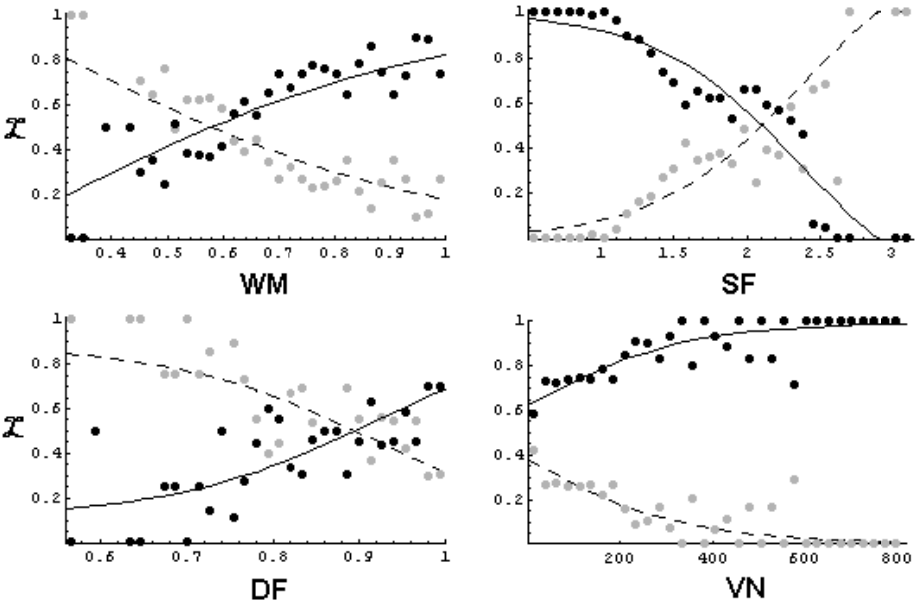


Fig. 1. Likelihood functions obtained by Multiple Sclerosis dataset. Grey and black points indicate respectively normal brain tissues ($\mathcal{L}_{NBT}$) and white matter potential lesions ($\mathcal{L}_{WMPL}$). Approximations by sigmoid functions are respectively indicated by dashed and continuous lines.

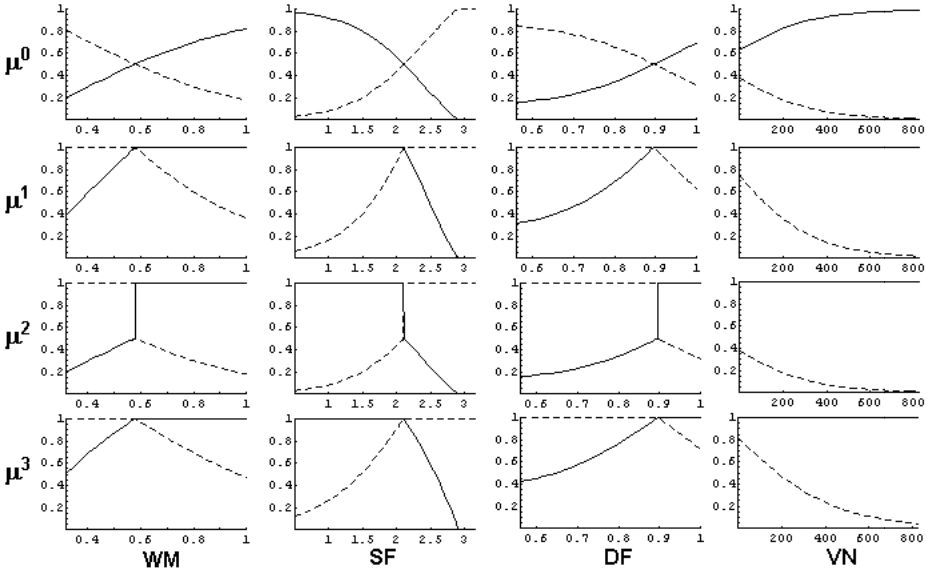


Fig. 2. Fuzzy sets obtained by likelihood functions, by using method 0, 1, 2 and 3. Dashed and continuous lines indicate respectively normal brain tissues (μ_{NBT}) and white matter potential lesions (μ_{WMPL}) fuzzy sets.

For each of the resulting sets of values, relative frequencies were calculated, by partitioning features into sub-intervals and dividing number of occurrences of a class for the total number of occurrences within each interval, as shown in figure 1. Coherently with relative frequencies, empirical likelihood functions $\mathcal{L}_{NBT}(x)$ and $\mathcal{L}_{WMPL}(x)$ were found, as shown in figure 1 with smooth lines, using sigmoid functions.

All methods described in section 3 were applied to likelihood functions, in order to identify, for each variable, the membership functions μ_{NBT} and μ_{WMPL}. In figure 2, a feature is considered on each column, and the application of different methods is reported on different rows. Each plot shows both of the fuzzy sets obtained.

By analyzing this figure and equations (12), (15), (16) and (17), characteristics of different approaches can be summarized as follows.

As far as probability-possibility transformations are concerned:

- method 0 gives probabilities, while the others transform them into possibilities;
- methods 1, 2 and 3 use normal transformations (for each value of the variable, there is at least one fuzzy set whose membership grade equals 1);
- method 1 and 2 use consistent transformations (at each point, the sum of probabilities of different classes do not exceed the maximal possibility); in case of only two classes, method 3 uses consistent transformation as well;
- method 2 uses the most specific transformation (μ^2 is less than μ^1 and μ^3);
- all transformations give possibilities which preserve probability order.

As far as the characteristics of the resulting fuzzy sets are concerned:

- the shape of the membership functions is strongly related to the approximation made for the likelihood functions;
- bounded supports were chosen here, between extreme values of each variable, but different choices can be done;
- in general, sub-normal fuzzy sets are obtained by all methods (see $\mu_{NBT}(VN)$);
- method 0 gives orthogonal fuzzy sets, while other methods do not;
- if a continuous and derivable function is chosen to approximate likelihood functions, in general, methods 0, 1 and 3 give continuous membership functions, while those of method 2 are discontinuous;
- all fuzzy sets can be interpreted with a likelihood view, that constructed by using method 0 in terms of probability, the others in terms of possibility.

Since a likelihood view interpretation is considered here, none of the methods give normal fuzzy sets, since this property should be required only in case of random set view interpretation of fuzzy sets.

If a probabilistic interpretation is considered, the method 0 is the only one which can be used; coherently, it gives orthogonal sets (the probability of any class, i.e. the sum of the two probabilities, must be 1). On the contrary, if a possibilistic interpretation is considered, methods 1, 2 and 3 can be used; coherently, at each point a fuzzy set ever exists whose membership is 1 (the possibility of any class, i.e. the maximum of the two possibilities, is 1).

Among possibility-based methods, method 2 is the most informative (specific). However, its application in this approach gives discontinuous membership functions, which are not intuitive and therefore scarcely interpretable.

Methods 1 and 3 give very similar results, but the first one is preferable because is a bit more informative. Moreover, it requires very less computational effort. The use of method 1 is mandatory when more than two classes exist, hence method 3 is not ever consistent.

In order to choose the most useful method for clinical DSSs uses, notice that DSSs need fuzzy sets interpretable with the likelihood view, and that in clinical fields the degree of possibility of a certain conclusion is required instead of its probability. Therefore, method 1 should be used. Other DSSs which require responses in terms of probability should use method 0.

5 Conclusions

Building DSSs based on fuzzy logic by means of statistical methods was previously discussed in literature. Starting from knowledge in the form of likelihood functions, their transformation into fuzzy sets is a required step to accomplish. However, to the best of our knowledge, only one method exists, which indentifies membership functions of fuzzy sets with the likelihood functions. Therefore, such a fuzzy set can be interpreted in terms of probability, rather than being a measure of vagueness.

However, vague concepts are better described by possibility measure. In order to obtain fuzzy sets interpretable in terms of possibility, an approach is given here. This approach considers the family of likelihood functions as, at each point, a probability distribution, and then suggests to transform these probabilities into possibilities. Hence, it can be composed with three different (most representative) probability-possibility transformations.

Therefore, by means of the proposed approach, three possibility-based methods arise, which are presented here together with a comparison among them and with the existing method.

While for likelihood view interpretation and probabilistic model the choice of method to use is straightforward, the best method is chosen here in order to obtain likelihood view interpretation, possibilistic model, continuous membership function, and order preserving, consistent and most informative probability-possibility transformation.

Acknowledgement. The authors are deeply grateful to the Department of Bio-Morphological and Functional Sciences of the University of Naples “Federico II” for providing them with the dataset pertaining the classification of Multiple Sclerosis lesions.

References

1. Zadeh, L.: Fuzzy sets. Inform. Control. 8, 338–353 (1965)
2. d'Acierno, A., De Pietro, G., Esposito, M.: Data driven generation of fuzzy systems: An application to breast cancer detection. In: Proc. of CIBB (2010)
3. Pota, M., Esposito, M., De Pietro, G.: Transformation of probability distribution into a fuzzy set interpretable with likelihood view. In: IEEE 11th International Conference on Hibrid Intelligent Systems (HIS 2011), Malacca, Malaysia, pp. 91–96 (2011)
4. Dubois, D., Prade, H.: Fuzzy sets and statistical data. European Journal of Operational Research 25, 345–356 (1986)
5. Bilgic, T., Türksen, I.B.: Measurement of membership functions: Theoretical and empirical work. In: Dubois, D., Prade, H. (eds.) Handbook of fuzzy sets and systems. Fundamentals of fuzzy sets, vol. 1, pp. 195–232. Kluwer, Dordrecht
6. Dubois, D., Prade, H.: Fuzzy sets and probability: Misunderstandings, bridges and gaps. In: Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, USA, vol. 2, pp. 1059–1068 (1993)
7. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems 1, 3–28 (1978)
8. Dubois, D., Foulloy, L., Mauris, G., Prade, H.: Probability-Possibility transformations, triangular fuzzy sets, and probabilistic inequalities. Reliable Computing 10, 273–297 (2004)
9. Yager, R., Kreinovich, V.: Entropy conserving probability transforms and the entaiment principle. Technical Report, MII-2518, Machine Intelligence Institute, Iona College, New Rochelle, NY (2004)
10. Klir, G.J.: A principle of uncertainty and information invariance. Int. Journal of General Systems 17, 249–275 (1990)
11. Dubois, D., Prade, H.: On several representations of uncertain body of evidence. In: Gupta, M.M., Sanchez, E. (eds.) Fuzzy Informatics and Decision Processes, pp. 167–181. North-Holland Pub. (1982)
12. Geer, J.F., Klir, G.: A mathematical analysis of information-preserving transformations between probabilistic and possibilistic formulations of uncertainty. Int. Journal of General Systems 20, 361–377 (1992)
13. Shafer, G.: A mathematical theory of evidence. Princeton Univerasity Press, NJ (1976)
14. Dubois, D., Prade, H.: Fuzzy sets and systems: Theory and applicatios. Academic Press, New York (1980)
15. Yamada, K.: Probability-Possibility transformation based on evidence theory. In: Joint 9th IFSA World Congress and 20th NAIPS International Conference, pp. 70–75 (2001)
16. Pota, M., Esposito, M., De Pietro, G.: Properties Evaluation of an Approach Based on Probability-Possibility Transformation. In: International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE 2011), December 3-12 (2011) (in press)

Combining Metaheuristic Algorithms to Solve a Scheduling Problem

Mª Belén Vaquerizo[1], Bruno Baruque[1], and Emilio Corchado[2]

[1] Computer Languages and Systems Area, University of Burgos, Spain
{belvagar,bbaruque}@ubu.es
[2] Depart. de Informática y Automática, University of Salamanca, Spain
escorchado@usal.es

Abstract. The Labour Scheduling problem in the context of any transport company is a complex optimization problem of that belongs to the class of NP-Hard problems. In these cases, it is not viable to try to find an exact solution and therefore, they require methods that assure the optimal management of the available resources in the tracing of the work calendars under the most suitable criteria of economy of times and costs. The main purpose of this research is to propose an efficient method to determine optimal shifts in a generic transport company, using bio-inspired methods. This method employs a two-step approach to obtain a solution. In a first stage, a Grasp algorithm is used to generate a viable solution. Then in a second stage, this preliminary solution is tuned, in order to obtain an optimal one, by using a Scatter Search algorithm.

Keywords: Bus Driver Scheduling Problem, Evolutionary Algorithm, N-P Hard Problems, Grasp Algorithm, Scatter Search Algorithm.

1 Introduction

Personnel scheduling is the problem of assigning staff members to shifts or duties over a scheduling period (typically a week or a month), so that certain constraints, organizational and personnel related, are satisfied [1].

This process normally consists of two different stages:

1. Staffing: Estimation of the number of drivers needed to cover the needs of working hours.
2. Scheduling: Construction of calendars of work to cover the estimation of drivers obtained in the stage of the staffing.

This study presents a method for solving this problem considering the limited time-availability of workers, among other constraints. Using the particular case of the organization of a transport company, the final objective of this work is the generation of calendars of shifts of work for drivers of a generic company, so the method used should be as flexible and adaptable as possible for any change or incident. These type

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 381–391, 2012.

of scheduling problems have been previously solved by means of metaheuristics algorithms, and among them, the Scatter Search algorithm [2, 3]. This study considers also a Scatter Search Algorithm to solve the problem while it also requires to adapt the algorithm to the characteristics of the problem to be solved. The remaining of this study is structured as follows: Section 2 summarizes previous efforts towards the solution of the presented problem. Section 3 describes the analysis of the problem to be solved. In Section 4, the method proposed for its resolution is presented. Section 5 includes the analysis of the results obtained. And, finally, Section 6 describes the conclusions and the future lines for this problem.

2 Previous Work

There are many different research works in the literature for solving the bus driver scheduling problem. Some of them [4, 5] use the set-covering formulation and Grasp, Simulated Annealing, Tabu Search or Genetic Algorithms to achieve a solution. Others [6, 7] present new mathematical models that represent all the complexity of the problem; considering in this way, different mathematical programming formulations for it.

Other works [8, 9, 10] apply different partitioning methods and set partitioning/ covering model, one after the other, to solve huge instances at once. A common approach to deal with these huge instances is to split them into several smaller ones. Others [11] can solve huge instances without splitting, and combining Lagrangian heuristics, column generation and fixing techniques.

Some [12] consider a set of trips to be covered, and the goal consists in finding a driver-vehicle schedule that serves the maximum workload and optimizes several economic objectives while satisfying a set of imperative constraints. Other [13] propose a hybrid solution method, using an iterative heuristic to derive a series of small refined sub-problem instances fed into an existing efficient set covering/ partitioning ILP model.

Other [2, 3] propose a Scatter Search Algorithm to solve this type of problem, because this algorithm is competitive and superior to other algorithms on most instances, especially in large-sized problems.

This work differs from the works examined in the literature in the way of constructing the final calendar, which minimizes the cost of the shifts programmed in the temporary horizon previously considered and satisfies the required demand. To do so, in a first step the construction of a preliminary solution is performed in a guided way by means of two constructive algorithms, obtaining a good starting point and satisfying a set of hard constraints. Later on, the Scatter Search Algorithm is applied to this initial solution, in order to obtain the final solution by improving the preliminary one.

The real number of resources is considered from the beginning, so the constructive algorithms can include the satisfaction of the major number of restrictions as a main goal. In this way, the set of solutions on which will be managed by the Scatter Search algorithm is constructed in guided way with two constructive algorithms, observing in

the first phase all the hard restrictions and in the second phase the major possible number of soft restrictions. This does not affect significantly the component of diversification of the Scatter Search algorithm trying to improve the final solution.

To the knowledge of the authors, none of the previously presented methods include this way of generating a solution: first generating a valid one and then refining it, simplifying significantly the second phase, as non-valid solutions will no longer be considered in this last one. In the rest of methods detailed, non-valid solutions can be generated during the functioning of the evolutionary models; that have to be discarded later on. In this sense, the proposed solution tries to favour the intensification over the solution, trying to improve a valid one; rather than the diversification over it, trying to obtain a novel valid solution.

3 Analysis of the Problem to Solve

Bus driver scheduling problem is one of the most important planning decision problems that transportation companies must solve and that appear as an extremely complex part of the general transportation planning system [14].

The objective is to have a method to be used as an automatic tool to produce driver schedules, generating real and useful schedules that can be implemented without many manual adjustments or modifications. For this purpose, the formation of these shifts must consider a group of rules that are specific to each organization. These rules are usually derived from other national and local regulations, being obligatory or not.

Typically, there are constraints in the total time worked, in the total extension of the shift (duration between the start and the end of the shift), etc. So the problem involves several constraints related to labour and company rules and therefore can also present different evaluation criteria and objectives.

On the other hand, it must consider three different aspects: it should try to maximize the satisfaction of the drivers (Labour Agreement), optimizing at the same time the resources of personnel of the companies, by minimizing their costs (Personal Costs), and trying to cover the maximum demand required (Demand of Work).

3.1 Information Requirements

The information obtained about the needs of the real-life problem provides the initial parameters for the creation of the calendar. First, it is necessary to establish the main restrictions to be considered in this problem [13, 12].

So, there are various rules, preferences and requests to comply with when allocating shifts. These constraints on the problem can be divided into two groups. In the category of hard constraints there are those that must always be satisfied. Some of the hard constraints considered in this work are: shift type requirements, maximum number of assignments, maximum number of consecutive days, etc.

In contrast, it is also possible to consider a high number of soft constraints on the personnel schedules. This kind of soft constraints must be preferably satisfied, but violations can be accepted to a certain extent. It is highly exceptional in practice to

find a schedule that satisfies all the soft constraints. These constraints have an associated penalty if not are satisfied, and this value would be added to the final value of the objective function for the solution obtained. Some of the soft constraints considered in this work are: minimum number of consecutive days, maximum number of consecutive free days, minimum number of consecutive free days, etc.

3.2 Driver Scheduling Problem Modelling

In this study, this is considered a real-life problem, with multiple pieces of information as data entries and multiple restrictions to satisfy. The solution must satisfy two opposite points of view: that of the company and that of its employees. The goal of the company is generating the work calendars with the minimum cost possible, whereas the goal of the worker is to obtain the major level of service possible respecting the highest number of soft restrictions as possible (which include their collective agreement or work conditions). Besides, an equitable distribution of the calendar of work with the rest of partners in the distribution of shifts of work is necessary. This requires achieving a level of commitment between both opposite goals.

For the transportation company, every driver has assigned a fixed cost per month. The main purpose of this study is to minimize the cost of the shifts programmed in the temporary horizon considered (day, week, month, etc.), with the restriction of having enough number of drivers in all periods of times to can satisfy the demand required.

In the first stages of the solution, this problem is formulated as a minimization problem whose objective is to determine the minimum number of driver shifts necessary to cover the estimated demand in each line, subject to a variety of rules and regulations that must be enforced. Once a minimum number of shifts is calculated, the problem changes in order to determine the estimated cost of each of the potential solutions.

Finally, the satisfaction of the drivers must be maximized by means of an equitable assignment of working hours, and considering their preferences (normally they are considered as soft constraints).

So, trying to satisfy the maximum required demand, the mathematic model [15] considered in this study is presented as follows:

$$Min \sum_{j=1}^{m} c_j x_j \tag{1}$$

where

$$\sum_{j=1}^{m} a_{j,i} x_j \geq r_i \qquad x_j \geq 0;\ x_j \in \mathrm{Z} \qquad (i=1\ldots h)$$

h = Number of periods of time considered (normally hours)

m = Number of shifts allowed or possible shifts

$$a_{j,i} = \begin{cases} 1 & \text{if the period i is included in the shift j} \\ 0 & \text{otherwise} \end{cases} \qquad \begin{matrix} (i = 1 \ldots h) \\ (j = 1 \ldots m) \end{matrix}$$

c_j = Cost of having a driver working during the shift j ; $(j = 1 \ldots m)$

r_i = Level needed of drivers in the period i ; $(i = 1 \ldots h)$

x_j = Number of drivers being employed at the shift j ; $(j = 1 \ldots m)$

To obtain the total cost of the solution considered, it is necessary to add the penalties for the soft restrictions not covered to the value obtained by this objective function.

In computational complexity theory, this type of problem belongs to the class of NP-complete problems [16]. Thus, it is assumed that there is no efficient algorithm for solving it and, in the worst case, the running time for an algorithm for this problem depends exponentially on the number of drivers, categories, shifts and so on, so that even some instances with only dozens of drivers cannot be solved exactly.

4 Proposed Method

The idea of this research is to construct an initial solution in a guided way, considering all the input information available and all the hard and soft restrictions previously defined. Once this initial process is finished, the solution obtained is evaluated to test if it is a feasible solution to try to improve it on a subsequent phase.

Three different sources of input information of are considered: the Labour Agreement, the Personnel Costs, and the Demand of Work to be satisfied; being the last one the most important. Once these three sets of data are defined, the satisfaction of the drivers must be maximized by means of an equitable assignment of working hours, and considering their preferences (normally they are considered as soft constraints). The two phases of construction of the calendar are initially applied, and the yielded solution is hopefully improved through a Scatter Search Algorithm. This last one is selected because the Scatter Search framework is one of the most flexible algorithms, allowing the development of alternative implementations with varying degrees of sophistication.

According to this, Fig. 1 shows the process of obtaining the final solution.

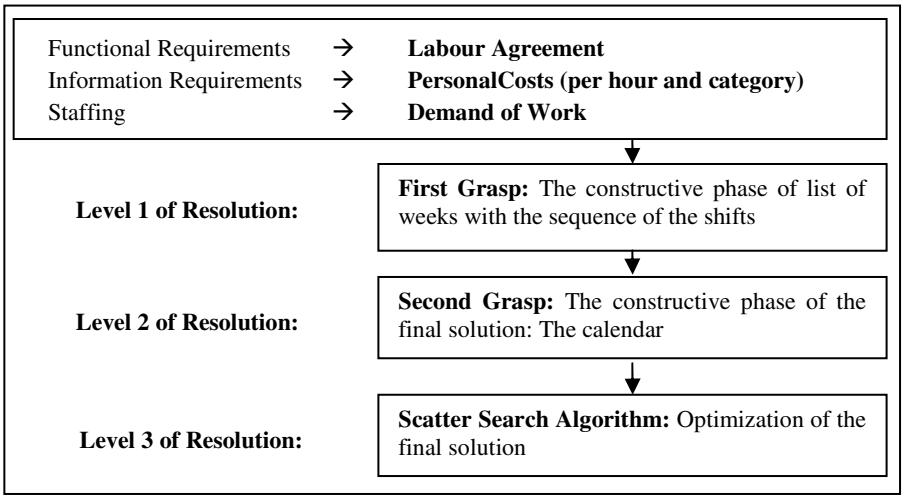


Fig. 1. The Process of obtaining the final Solution with two Constructive Algorithms and a Scatter Search Algorithm

The construction of the solution consists of three different phases:

An Initial Constructive Phase Obtaining a List of Weeks Satisfying the Conditions Related to the Sequence of Shifts and the Hard Constraints

As a first step, it is necessary to generate the list of all the possible weeks that can be obtained for a single driver. This is obtained by using a standard Grasp (greedy randomized adaptive search procedure) algorithm.

As the output is composed of a high number of elements, a subsequent step must be carried out to reduce the size of its elements, using a first group of restrictions. This includes all the hard restrictions that are selected by the user at the beginning of the execution.

The next step is to examine which weeks fulfil the sequence of shifts conditions to be joined with other previously given, without violating some of the previous restrictions. This will save computation time in a later phase.

The size chosen for the reduced list is the number of drivers working for the company studied.

A Subsequent Constructive Phase (from the Results Obtained in the Previous Phase), Considering the Soft Constraints and Obtaining the Final Solution: A Valid Calendar

This stage is also composed by two different parts, again implemented by means of a Grasp algorithm. The first one is the constructive phase of the calendar. In this phase, an initial group of n weeks is extracted randomly from the results of the previous main step. In the second stage, a post-processing phase is performed over the result obtained in this way in the previous part, examining if the list chosen fulfils a second block of restrictions composed of the soft restrictions determined by the user. In case

that these restrictions are satisfied, such list is stored. Otherwise, one of the weeks that form this solution is rejected, substituting it by other one obtained from the post-processing phase. Again, it is analysed to check if it fulfils the requirements. This way, possible combinations are generated and rejected in an iterative way.

Finally, a study of improvement of the obtained solution is carried out, looking for some other combinations in the environment of the solution that can improve the solution considered.

Finally, the Optimization of the Preliminary Solution is Constructed Using a Scatter Search Algorithm

Scatter Search is a population and evolutionary method. In contrast to other evolutionary methods like genetic algorithms, scatter search is mostly based on systematic designs and methods with the purpose of creating new solutions. It uses strategies for search diversification and intensification that have proved effective in a variety of optimization problems.

This study makes use of this algorithm, because it has demonstrated to solve efficiently this type of complex problem, even for instances of the problem of great size. It operates with a set of reference of the population. In SS it is possible to generate a very significant number of combinations with few individuals. SS systematicly introduces diversity in the set reference.

To apply a Scatter Algorithm [14, 17, 18] to this Scheduling problem, first the problem must be represented as a genome. To make sure that a certain genome is a feasible solution, it must be checked if it obeys the precedence constraints previously indicated. The fitness function of a solution considers the number of penalties and the value of each one of the restrictions previously considered, as well as the number of restrictions not met by this solution.

The algorithm proceeds to initialize a population of solutions considering the solution generated by the second constructive phase as an entry. All the solutions in the population are generated from that initial solution, and the evolution happens in generations.

In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected in a guided way from the current population (based on their fitness), and modified (recombined and possibly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Thus, each generation consists on updating the population of individuals, using for it a new population obtained by recombination of the characteristics of some selected individuals.

The algorithm ends when, either a maximum number of generations have been produced, or a satisfactory fitness level has been reached for the population. Among the different final solutions obtained, the solution that is closer to the ideal value or that is in the environment of the ideal solution will be selected.

5 Analysis of Results

This section includes a description of a very simple generic problem, along with the solution found. The calendar generated in the application for every driver is a succession of days, and the temporary horizon can be a week, a month, a semester or a year.

The constructive phases have been used to identify the lists of feasible weeks, yielding a list of shifts for each driver. Afterwards, the final annual calendar has been constructed from these feasible lists of shifts in the 4 weeks.

The following is only an explanatory example, considering only 4 drivers. For every day of the week, working place and shift, a cost is defined by hour and a cost of overtime. The restrictions considered in this example are showed in Table 1.

Table 1. Lists of hard and soft restrictions to be considered

Hard Restrictions	1. The shifts of work to be considered: 2 (M, A) 2. The hourly ranges indicated in the shifts. The hourly ranges cannot be overlapped between shifts: 8 hours per shift 3. Types of days of work considered (TC: complete time, TP: partial time…): Only TC. 4. The number of drivers: 4 5. Different categories in which the drivers are grouped: Only one category. 6. Different groups of days defined: Labour days, weekly days of rest and days of annual vacations 7. Maximum number of consecutive days: 6
Soft Restrictions	1. Maximum number of days worked in a year: 247. 2. Holidays and days off in the calendar of work: 1st January, 25th December (complete day, all the shifts), 24th and 31st December (only night shifts). 3. Preferences of the drivers to not to be employed at a certain shift: Driver 4 prefers not to work in Afternoon Shift. 4. Restrictions about labour days, shift and days of rest: All drivers prefer not to work on Sunday 5. Maximum number of a shift type per week: 1 6. Maximum number of assignments per day of the week: 6 7. Maximum number of assignments for each shift type: 6 8. Maximum number of consecutive working weekends: 2

Considering a section composed by only 4 drivers, the cycle of this section or department will be constituted by 4 weeks.

In the results showed next; the temporary horizon has been considered as a year, only two shifts have been considered (M = Morning, A = Afternoon, B = Break), and the schedule includes only 4 drivers. To obtain the annual solution, is necessary to extend the 4 solutions obtained until this moment to the 52 weeks of the year. The final solution satisfies all of the hard restrictions and most of soft restrictions, with the exception the one marked with 3 on Table 1.

The list of shifts to complete in 4 weeks for each of the 4 drivers considered is shown in Table 2.

Table 2. Lists of shifts to be worked by 4 drivers in 4 weeks

	WEEK 1							WEEK 2						
Driver 1	M	M	M	M	M	B	B	A	A	A	A	A	A	B
Driver 2	A	A	A	A	A	A	B	B	M	M	M	M	M	B
Driver 3	B	M	M	M	M	M	B	A	A	A	A	A	B	B
Driver 4	A	A	A	A	A	B	B	M	M	M	M	M	B	B
	WEEK 3							WEEK 4						
Driver 1	B	M	M	M	M	M	B	A	A	A	A	A	B	B
Driver 2	A	A	A	A	A	B	B	M	M	M	M	M	B	B
Driver 3	M	M	M	M	M	B	B	A	A	A	A	A	A	B
Driver 4	A	A	A	A	A	A	B	B	M	M	M	M	M	B

The solution obtained satisfies all the hard constraints and 89% of the soft constraints previously considered to generate the calendar of work; such as labour days, shift and days of rest, and some restrictions about intervals of the vacations and about days of local or national holiday.

The final solution considers the extension of these lists through the 52 weeks in the year. As Table 3 shows, the final results include a very similar number of worked days and shifts in a year for each driver, applying this pattern of solution.

Table 3. Final results of the number of worked days in a year

Drivers\Shifts	**Morning**	**Afternoon**	**Total**
Driver 1	130	143	273
Driver 2	129	144	273
Driver 3	128	145	273
Driver 4	133	140	273

As it can be observed in Table 3, the differences between the drivers are not relevant, and taking into account that the rest of restrictions were satisfied, it can be concluded that the application of the constructive method and optimization algorithm previously described are considered as satisfactory to solve this problem. Both the requirements of the company (minimization of the costs to cover all routes) and those of the workers (almost no differences in working hours and most of other constraints covered) have been observed.

For this solution the execution time is 4915 ms, the best solution has a cost of 138350 and the worst solution has a cost of 273725.

As additional remark, it has been observed that if the number of hard restrictions increases, the execution time reduces and the obtained solutions deteriorate significantly. This is due to the fact that there are few possibilities of finding solutions that satisfy many hard restrictions (normally if there are a lot of hard restrictions they have opposite interests). This makes the execution to stop sooner, as no improvements

are made during a determined number of iterations. On the contrary, if the number of hard restrictions decreases, the time of execution increases and the obtained solutions are better.

Based on results obtained with greater instances, with this way of solving this problem of driver scheduling when the size of the problem increases, advantages of using this method become clearer, as it obtains optimal results without the computational complexity of other methods, which have to manage invalid solutions.

6 Conclusions

In this work, the Bus Driver Scheduling Problem, which is an important aspect of the Transportation Planning System, has been analysed. The presented model for obtaining the final calendar has been designed trying to achieve simplicity, solution quality and applicability, as its main characteristics.

Its main characteristic is the partition of the main problem of calendar construction in three different stages. That way, first an initial solution is constructed and then the solution is improved to obtain the best solution possible. This guided approach avoids the pitfalls of other, more random, solutions; in which more computing time to get to a correct solution would be higher. It also favours the construction of solutions that comply with the majority of the restrictions that the problems require. As results prove, this method ensures that all hard restrictions are satisfied, while it helps to fulfil also with the majority of the soft ones.

Future lines of work include the consideration of extending its use to another Scheduling Problems, the consideration of others constraints, and the consideration of uncertain data using the Fuzzy Sets Theory to solve the Scheduling Problem.

Acknowledgments. This research has been partially supported through the projects of the Spanish Ministry of Science and Innovation CIT-020000-2009-12 and TIN2010-21272-C02-01 (funded by the European Regional Development Fund).

References

1. Brusco, M.J., Jacobs, L.W.: Personnel tour scheduling when starting time restrictions are present. Management Science 44, 534–547 (1998)
2. Rezanov, N., Ryan, D.: The train driver recovery problem. A set partitioning based model and solution method. Computers & Operations Research 37(5), 845–856 (2010)
3. Tang, J., Zhang, J., Pan, Z.: A Scatter Search Algorithm for solving vehicle routing problem with loading cost. Expert Systems with Applications 37(6), 4073–4083 (2010)
4. Portugal, R., Ramalhingo-Lourenco, H., Paixao, J.P.: Driver Scheduling Problem Modelling. Public Transp. 1, 103–120 (2009)
5. Ramalhingo-Lourenco, H.: The Crew-Scheduling Module in the GIST System. UPF Economics Working Paper No. 547 (2001)
6. Carraresi, P., Gallo, G., Rousseau, J.M.: Relaxation approaches to large scale bus driver scheduling problems. Transportation Research Part B: Methodological 16(5), 383–397 (1982)

7. Mesquita, M., Paias, A.: Set Partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem. Computers & Operations Research 35(5), 1562–1575 (2008)
8. Abbink, E.J.W.: Solving large scale crew scheduling problems by using iterative partitioning. In: 7th Workshop on Algorithmic Methods and Models for Optimization of Railways, ATMOS 2007 (2007)
9. Laurent, B., Hao, J.-K.: Simultaneous Vehicle and Crew Scheduling for Extra Urban Transports. Computers & Industrial Engineering 53(3), 542–558 (2007)
10. Ramalhingo-Lourenco, H., Portugal, R.: Metaheuristics for The Bus-Driver Scheduling Problem. Economic Working Papers Series, vol. 304. Universitat Pompeu Fabra (1998)
11. Abbink, E.J.W., et al.: Solving Large Scale Crew Scheduling Problems in Practice. Research Paper. Report/Econometric Institute, Erasmus University Rotterdam, pp. 1-19. Erasmus School of Economics (ESE). This publication is part of collection Econometric Institute Research Papers Published by Econometric Institute (2010)
12. Kwan, R., Kwan, A.: ASK Effective search space control for large and/or complex driver scheduling problems. Annals of Operations Research 155(1), 417–435 (2007)
13. Tavakkoli-Moghaddam, R., Makui, A., Mazloomi, Z.: A new integrated mathematical model for a bi-objective scatter search algorithm. Journal of Manufacturing Systems 29 (2-3), 111–119 (2010)
14. De Leone, R., Festa, P., Marchitto, E.: The Bus Driver Scheduling Problem: a new mathematical model and a GRASP approximate solution. Tech. Rep. 22, University of Napoli, Federico II (2006)
15. Danzing, G.B.: A Comment on Edieś 'Traffic Delays at Toll Boothś. Operations Research 2(3), 339–341 (1954)
16. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman (1979)
17. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109–133 (1995)
18. Glover, F., Laguna, M., Martí, R.: Scatter Search. In Advances in Evolutionary Computation: Theory and Applications, pp. 519–537. Springer, Heidelberg (2003)

Image Analysis Pipeline for Automatic Karyotyping

Izaro Goienetxea[2], Iñigo Barandiaran[1], Carlos Jauquicoa[1], Grégory Maclair[1], and Manuel Graña[2]

[1] Vicomtech
[2] Dpto. De Ciencias de la Computación e Inteligencia Artificial, UPV-EHU
http://www.vicomtech.org

Abstract. The karyotyping step is essential in the genetic diagnosis process, since it allows the genetician to see and interpret patient's chromosomes. Today, this step of karyotyping is a time-cost procedure, especially the part that consists in segmenting and classifying the chromosomes by pairs. This paper presents an image analysis pipeline of banded human chromosomes for automated karyotyping. The proposed pipeline is composed of three different stages: an image segmentation step, a feature extraction procedure and a final pattern classification task. Two different approaches for the final classification stage were studied, and different classifiers were compared. The obtained results shows that Random Forest classifier combined with a two step classification approach can be considered as an efficient and accurate method for karyotyping.

Keywords: Image Analysis, Kariotyping, Classifier Ensembles.

1 Introduction

Genetic diagnostic has rapidly evolved in recent years thanks to the increase of computation resources, allowing to increase the working resolution from where anomalies can be detected, and also reducing computation time. The study of human chromosome is important because changes in the number or the structure of chromosomes may lead to congenital anomalies or cancer. Chromosomes are complex structures that are located in the nucleus of cells mainly compound of DNA and other proteins, as shown in figure 1.

Structural changes can not be globally studied by any molecular biology technique. Karyotyping is still the most common technique used for chromosome structural analysis. This technique plays an important role in many clinical studies. Karyotyping technique tries to describe the number and shape of the chromosomes inside the cells. The chromosomes are split and depicted in an ideogram, ordered by their appearance or morphology (position of centromere) and their size. The karyotype is also used as the complete set of chromosomes in a species or an individual organism.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 392–403, 2012.

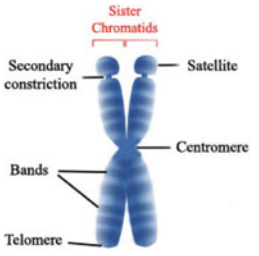


Fig. 1. Structure of a Chromosome

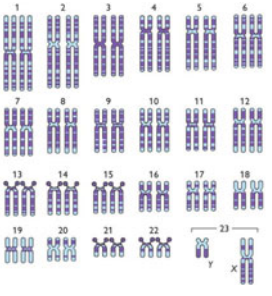


Fig. 2. Karyotype ideogram

In order to be able to identify and to classify chromosomes, the cells are processed during the metaphase stage. In this step of the cellular division the chromosomes are better seen due the DNA duplication and the two chromatids are already formed. Once the cells are in this stage, giemsa stain is applied in order to be able to visualize the chromosome bands. Depending on the length, position of centromere and the pattern of bands, the chromosomes are classified and rearranged as shown in figure 2.

Nowadays there exists software for karyotyping that extracts chromosomes from microscope images of cellular nucleous in metaphase state, classifying them afterwards. This process is being more and more automated, but still some processes need human intervention, thus augmenting time for processing, and being prone to error.

The rest of the article is organized as follows: in section 2 an small review of approaches and techniques related with karyotyping is shown. Next, in section 3an implementation of our proposed pipeline is depicted, following with some results in section 4. Finally, some conclusions and future work are described in section 5.

2 State of the Art

2.1 Feature Extraction

The goal of feature extraction is finding a small parameter set that best describes each class. These features can be separated in two groups: morphological-based features and texture-based features. Morphological-based features provide information about the size and shape of the chromosome, but they are not enough to make a complete classification, that is why texture-based features are necessary. The most important morphological features are length, the medial axis and centromeric index (CI) [2,3,4]. Texture-based features describe band patterns of chromosomes. The most popular are density profile [4], mean gray value profile, gradient profile and shape profile [3]. These profiles are used to extract Fourier coefficients [5] or WDD (Weighted Density Distribution) coefficients [2,3], to be used in the classification.

2.2 Classification

For chromosome characterization or classification artificial neural networks (ANN) have been widely used [6,7]. Lerner [7] suggested that neural networks are the best classifier for this task, specially when the number of different classes is limited. Some approaches [8] and [9] are focused on the classification of only one group of chromosomes such as group E (16, 17 and 18), being very accurate. Similar approaches can be found in [10] where a wavelet based artificial neural network is applied for the classification of same group, obtaining better results.

For improving the accuracy of neural networks for chromosome classification, some authors propose to use a two step classification approach [3]. In the first step chromosomes are classified as one of the Denver groups. In the second step, final classification is carried out in every group separately. This approach has been also studied by using Bayesian Classifiers instead of neural networks obtaining similar results [11]. This work shows that the two step approach is much more accurate than single step approach.

Some other methods have been also applied for chromosome classification such as Fuzzy similarity relation [12]. In this work chromosomes are firstly divided in groups using fuzzy logic membership functions or relations. In a second step groups are rearranged depending on the centromere positions of their members. Finally, every chromosome is classified by correlation of their band pattern with a band pattern model of their corresponding group.

3 Material and Methods

In this section, we are going to detail the proposed image analysis pipeline, which as shown in figure 3 can be divided in three steps: the segmentation step, feature extraction step and classification step. Each one of these steps will be detailed in the following sections.

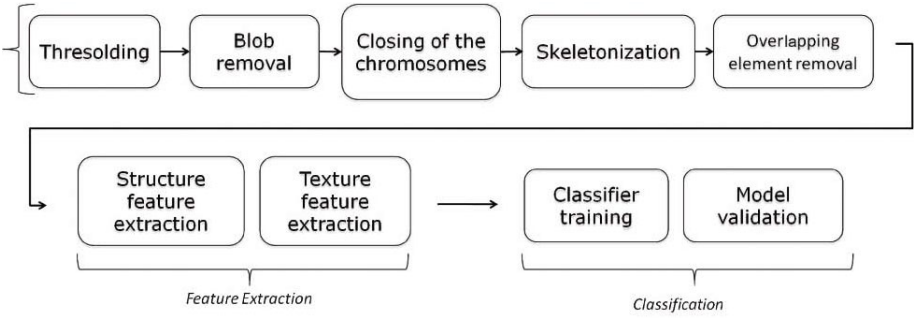


Fig. 3. Proposed Pipeline

3.1 Segmentation

As shown in figure 3, the first step of the segmentation procedure is the use of Otsu threshold algorithm. As a result, a binary image is obtained, in which the identified objects are both chromosomes and cell nucleus, see figure 4.

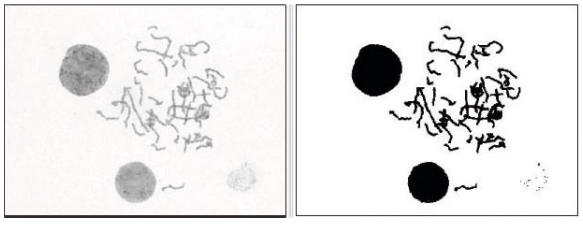


Fig. 4. Example of Otsu segmentation

In order to remove the cell nucleus from the analysis, a blob removal algorithm based on object size is used. To quit the eventual wholes in the chromosomes that can result of the Otsu thresholding, a closing is applied on the binary image objects.

To be able to detect overlapped chromosomes and proceed with the feature extraction, it is necessary to compute the chromosome skeleton. In this work, an erosion based implementation has been used, thinning the chromosome until getting a one-pixel-width object. Using these skeletons, overlapped chromosomes can be detected and removed from the study, looking for a pixel with more of two neighbors all along the skeleton axis.

3.2 Feature Extraction

In this work, we used morphological-based features and texture-based features as input for the classification. Morphological-based features has been demonstrated to be very important in the Denver group classification [3], so that the

centromeric index and the relative length of the chromosome have been chosen to be part of the parameters used to build our classification model. Texture-based features are needed to ensure the classification of the chromosomes within each Denver group, so that the density profile, the shape profile and the median intensity profile have been integrated in our classification model. For the extraction of several of these features, it is necessary to work on straight chromosomes. Thus, a straightening algorithm has been used on the chromosome images when it was required. In the following, we detailed the features selected:

Centromeric Index (CI): This feature expresses the ratio between the length of the short arm of the chromosome and the length total of the chromosome. The extraction of this feature is hard to automate due to the difficulty for finding automatically the centromere position. In this work, centromere positions have been located manually, with the help of the Vogel and Motulsky table[18].

Length: To compute the length L_n of the chromosome, the length of the axis (skeleton) of the chromosome is used. As each metaphase image used to do the karyotype is acquired at different moment of the metaphase process, the distribution and the size of the chromosomes are different from one image to an other. To deal with this inhomogeneity in the dataset, the normalization of chromosomes length is done:

$$L_n = \frac{L_r}{L_{max}} \tag{1}$$

,where L_r is the relative length of the chromosome, and L_{max} the length of the largest chromosome of the metaphase.

Density Profile: This profile characterizes the property of the chromosome-bands pattern. Each normalized value $d_N(i)$ of the profile results of the sum of the intensity of the pixels of the i^{th} transversal line weighted by the width of the chromosome at the i^{th} position of the chromosome skeleton. In the following equations, $d_w(i)$ is the non-normalized density profile value, m is the number of pixels of the perpendicular line to the skeleton, $d(i,j)$ is the pixel value at the coordinates (i,j), and $w(i)$ is the width of the chromosome at the i^{th} position:

$$d_N(i) = \frac{d_w(i) - d_{wMIN}(i)}{d_{wMAX}(i)} \quad (i = 0, 1, \ldots, n-1), \tag{2}$$

with,

$$d_w(i) = \frac{I_i}{w(i)} \quad (i = 0, 1, \ldots, n-1) \tag{3}$$

,and,

$$I_i = \sum_{j=0}^{m-1} d(i,j) \quad (i = 0, 1, \ldots, n-1) \tag{4}$$

The figure 5 shows a chromosome and its corresponding density profile.

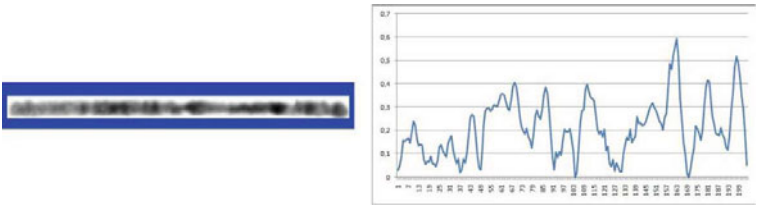


Fig. 5. Straighted chromosome and its corresponding density profile

Shape Profile: In order to obtain the shape profile, the values have been computed using, as for the density profile, the intensity of the pixels of each transversal line all along the skeleton. This profile characterizes the ratio between the second moment and the moment 0 of each transverse line, using the following equation:

$$s_w = \frac{\sum_{j=0}^{m-1}(d(i,j)dist(i,j)^2)}{\sum_{j=0}^{m-1} d(i,j)} \quad (i = 0, 1, \ldots, n-1), \tag{5}$$

with $d(i,j)$ the gray value of the pixel at the coordinates(i,j), and $dist(i,j)$ the Euclidean distance between the i^{th} position of the skeleton and the position of coordinates (i,j).

The figure 6 shows a chromosome and its corresponding shape profile.

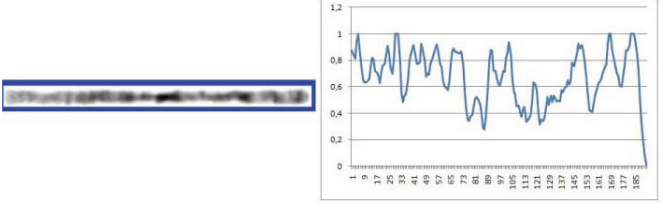


Fig. 6. Straighted chromosome and its corresponding shape profile.

Median Gray Intensity Profile: This profile measures the median intensity levels of the pixels of each transversal line along the chromosome skeleton using equation:

$$g_w = \frac{\sum_{j=0}^{m-1} d(i,j)}{n} \quad (i = 0, 1, \ldots, n-1), \tag{6}$$

with *d(i,j)* representing the intensity value of the pixel in *(i,j)* coordinates and n is the total number of pixels of the i^{th} transversal line of the chromosome .The figure 7 shows a chromosome and its corresponding median gray intensity profile.

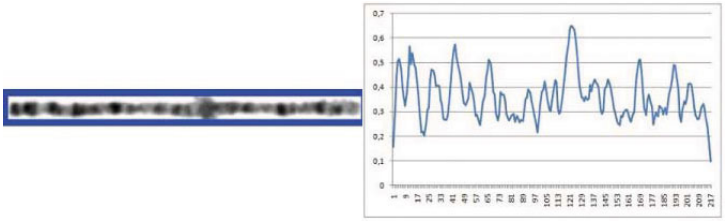


Fig. 7. Straighted chromosome and its corresponding intensity profile

In order to reduce the dimensionality of these profiles and to be able to use the information they contain for the classification, we decided to use the six first WDD coefficients for each profile. To extract the WDD coefficients, we firstly compute the WDD functions, and then, we multiplicate these functions with the profile from which we want to extract the coefficient. The WDD functions are obtained using the following equation:

$$w_n(x) = \left[2\left\lfloor n.\frac{2.x+1}{2.L}\right\rfloor + 1 - 2.n.\frac{2.x+1}{2.L}\right](-1)^{\left\lfloor\left(\frac{2.x+1}{2.L}\right)-1\right\rfloor} \quad 0 \leq x < L, \quad (7)$$

with $w_n(x)$ the n^{th} WDD function and L the profile length.

Once the WDD functions obtained, the coefficients are obtained using the equation:

$$WDD_n = \sum_{x=0}^{L-1} w_n(x)p(x) \quad 0 \leq x < L \quad (8)$$

Table 1 presents a summary of the chromosome features used in this work for the classification.

Table 1. Summary of chromosome features used for classification

Feature Number	Description
1	Centromeric Index (CI)
2	Relative length
3-8	Six density profile WDD coefficients
9-14	Six shape profile WDD coefficients
15-20	Six median gray intensity profile WDD coefficients

3.3 Classification

In this section we want to test different supervised classifiers in order to classify chromosomes. Chromosome classification or categorization is a basic step in the process of karyotyping. We have tested the following classifiers:

- **Bayesian Networks:** Also known as Bayes Network, belief networks, are probabilistic structures that captures or represents a set of random variables (features) of observed data and their conditional dependency, by using directed acyclic graphs. [13].
- **Multilayer Perceptron (MLP):** Is a feed forward artificial neural network classifier. An MLP consists of multiple layers of nodes in a directed graph, with each layer is fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes back propagation technique for training the network that allows the classifier to distinguish data that is not linearly separable [14].
- **J48 Decision tree:** This is an implementation of the popular Quinlan's [15] C4.5 algorithm for growing decision trees.
- **Random Forest:** A random forest ensemble (also known as random subspace) [16,19] uses a large number of individual, unpruned decision trees (weak learners) which are created by randomizing the split at each node of the decision tree construction. Each tree is likely to be less accurate that a tree created with exact splits. But by combing several of these "approximate" trees in an ensemble, the final accuracy is improved.

We have selected Bayesian Networks and Artificial neural networks classifiers because they have been widely applied for chromosome classification in recent years. In addition, we have used a C4.5 algorithm for constructing decision tree classifier, in order to be able to test against a tree based ensemble classifier such as Random Forest. This classifier has not been widely used for chromosome classification but is a popular ensemble technique in pattern recognition. It is well known that this type of classifiers shows great trade off between classification accuracy and computational resources.

All classifier implementations and tests were carried out used Weka open source software version 3.6[17]. Parameters of every classifier were set as their default values, not tunning any parameter of any classifier. It is clear that modifying the parameters of each classifier, such as number of trees in the forest in case of Random Forest, can improve the overall performance of every classifier.

4 Results

We have carried out two different approaches for chromosome classification. In the first approach, we have used a single classification method, where all chromosomes are directly classified. In another approach, we have used a two steps method where chromosomes are firstly classified in Denver groups, and then, a classification in every specific group is applied.

4.1 Single Step Classification

Tests for one step approach were carried out by using 1288 samples of 24 classes (1,2,...X,Y). One sample refers to one chromosome image. The set is composed of 56 samples of classes 1 to 22, 44 samples of class X and 12 samples of class Y.

Some approaches for dimension reduction or feature importance such as Correlated Feature Selection (CFS) has been applied in order to test how they affect the results. The CFS algorithm returns a set of features that best represents class attribute by using a correlation measure of mutual information between class attribute and the rest of attributes. In all cases, results were worse than the studies using all features without reduction or projection.

Results obtained with every classifier using the whole set of features is shown in table 2.

Table 2. Accuracy of different classifiers in Denver Group (one step)

Classifier	Median	Typical deviation
Multilayer Perceptron	0.8271	0.00546
Bayesian Network	0.8383	0.00629
J48	0.8261	0.00686
Random Forest	0.8672	0.00712

4.2 Two Steps Classification

This section shows the results obtained by applying a two step based approach, where chromosomes are classified in any of the Denver groups (Group A,Group B ... ,Group G) in a first step. In the second step a specified classification was applied to each group. Every test consist in a 10 times 10 fold cross validation procedure for each different classifier. The training/test set used in this approach is the same as the one used in the single step approach.

In this first step only morphological features, such as relative length and the CI has been used on a dataset compound of 1288 samples. Results are shown in table 3. We obtained a maximum accuracy of 0.9698 in case of Random Forest classifier.

Table 3. Accuracy of different classifiers in Denver Group (two steps)

Classifier	Median	Typical deviation
Multilayer Perceptron	0.9337	0.00294
Bayesian Network	0.962	0.00273
J48	0.9643	0.00171
Random Forest	0.9698	0.00261

In this study we have also applied CFS algorithm for feature importance evaluation or dimension reduction. In general, for every group we have selected the

five or six most relevant features selected by CFS. For example, for group A we used the relative length, CI, second and fourth density profile components, and third and fifth components of median density profile. In case of group B we used first and second density profile components, the first shape profile component, and the second and fifth of median density profile. The rest of groups share almost the same configuration, being the relative length and CI the most relevant features in almost every group. Tests with and without dimensional reduction show no significant differences in accuracy, so we kept the reduction in order to decrease computational time.

Table 4. Results for Denver Groups Classification

Clasificador	Group A	Group B	Group C	Group D	Group E	Group F	Group G
Multilayer Perceptron	0.9762	0.8857	0.806	0.888	0.9735	0.8901	0.8696
Bayesian Network	0.9726	0.8857	0.8128	0.8807	0.9783	0.8811	0.8978
J48	0.9595	0.8857	0.7702	0.8747	0.9711	0.8558	0.8739
Random Forest	0.9786	0.8982	0.822	0.9048	0.9856	0.8919	0.9043

It is clear that approaches based on two steps are significantly better than a single classification step. In the two steps scenario, our results shows that Random Forest obtains slightly better results compared with other classifiers such as Bayesian Networks. However, either training time and classification time are much lower in case of Random Forest compared to the rest of tested classifiers. Also, Random forest implicitly performs feature selection, and feature importance what can be seen as an extra benefit in several studies, such as the one presented in this article. This feature in addition to the possibility of parallel execution of classifier ensembles, turns Random Forest the most appropriate classifier for our purposes among tested ones, because we are interested in approaches with good trade-off between accuracy and efficiency. It worth notice the reduction of accuracy in Group C (83%) compared with the rest of groups (97%). We think that this accuracy reduction is due to the fact that group C is the biggest one (7 chromosomes), and that there is a closer relation between centromere position feature (CI) and class feature. As explained in section 3.2 the position of centromere is not very accurately extracted from images so this could have a negative impact in classification, more severe in case of group C.

5 Conclusions and Future Work

In this paper an image processing and analysis pipeline for automatic karyotyping of microscope images of metaphase is proposed. We have analyzed the feasibility of a system that employs 10 metaphase images of the same patient, in order to overcome the problem of overlapping information, typical in a single image scenarios. Results show that, in general, the two steps approach is more accurate than the single step approach. Related with classification, we have shown that an approach based on classifier ensemble can reach similar results

or better that Neural networks. More important, classifier ensemble are suitable for parallelization, what opens an important research line for optimizing computation time, thus allowing high throughput karyotyping. Even that results are promising, we still need to test with more patient in order to verify the benefits of the approach. Moreover, some improvements needs to be done such as the need for an automatic chromosome centromere detection. Moreover, the problem of overlapped chromosomes is an open issue, and is not completely solved by community, being necessary the human intervention for disambiguation.

Finally, we want to study several approaches and heuristics for improving the performance of classifier ensembles. These approaches are focused on the increase of the diversity in the ensemble. It is well known that increasing the diversity or increasing the negative correlation between members on the ensemble the final accuracy of the classifier can be significantly increased.

The authors would like to express their acknowledgments to Genetadi[1] for providing images that allowed the realization of this study.

References

1. Wang, X., Zheng, B., Wood, M., Li, S., Chen, W., Liu, H.: Development and evaluation of automated systems for detection and classification of banded chromosomes: current status and future perspectives. J. Phys. D: Appl. Phys. 38, 2536–2542 (2005)
2. Piper, J., Granum, E.: On Fully Automatic Feature Measurement for Banded Chromosome Classification. Cytometry 10, 242–255 (1989)
3. Ming, D., Tian, J.: Automatic Pattern Extraction and Classification for Chromosome Images. J. Infrared Milli Terahz Waves 31, 866–877 (2010)
4. Cho, J.: A Hierarchical Artificial Neural Network Model for Giemsa-Stained Human Chromosome Classification. In: IFMBE Proceedings Biomed 2006, pp. 12–15 (2007)
5. Ritter, G., Schreib, G.: Profile and feature extraction from chromosomes. In: ICPR, vol. 2, pp. 287–290 (2000)
6. Ritter, G., Pesch, C.: Polarity-free automatic classification of chromosomes. Computational Statistics & Data Analysis 35, 351–372 (2001)
7. Lerner, B.: Toward A Completely Automatic Neural-Network-Based Human Chromosome Analysis. IEEE transactions on systems, man, and cybernetics—part b: cybernetics 28(4) (1998)
8. Srisang, W., Jaroensutasinee, K., Jaroensutasinee, M.: Segmentation of Overlapping Chromosome Images Using Computational Geometry. Walailak J. Sci. & Tech. 3(2), 181–194 (2006)
9. El Emary, I.M.M.: On the Application of Artificial Neural Networks in Analyzing and Classifying the Human Chromosomes. Journal of Computer Science 2(1), 72–75 (2006)
10. Oskouei, B.C., Shanbehzadeh, J.: Chromosome Classification Based on Wavelet Neural Network, pp.605–610 (2010), doi:10.1109/DICTA.2010.107
11. Lerner, B., Guterman, H., Dinstein, I.: A Classification-Driven Partially Occluded Object Segmentation (CPOOS) Method with Application to Chromosome Analysis. IEEE Transactions On Signal Processing 46(10), 2841–2847 (1998)

[1] www.genetadi.com

12. Karslıgil, M.E., Karsligil, M.Y.: Fuzzy Similarity Relations for Chromosome Classification and Identification. In: Solina, F., Leonardis, A. (eds.) CAIP 1999. LNCS, vol. 1689, pp. 142–148. Springer, Heidelberg (1999)
13. Ben-Gal, I.: Bayesian Networks. Encyclopedia of Statistics in Quality & Reliability. Wiley & Sons (2007)
14. Noriega, L.: Multilayer Perceptron Tutorial. School of Computing. Staffordshire University (2005)
15. Quinlan, R.J.: Learning with Continuous Classes. In: 5th Australian Joint Conference on Artificial Intelligence, Singapore, pp. 343–348 (1992)
16. Breiman, L.: Random Forests. J. Machine Learning 45(1), 5–32 (2001)
17. http://www.cs.waikato.ac.nz/ml/weka/
18. Vogel, F., Motulsky, A.G.: Human genetics: problems and approaches ISBN 978-3-540-37653-8
19. Barandiaran, I., Paloc, C., Graña, M.: Real-time Optical Markerless Tracking for Augmented Reality Applications. Journal of Real-Time Image Processing 5(2), 129–138 (2010)

A Hybrid Gradient for n-Dimensional Images through Hyperspherical Coordinates

Ramón Moreno and Manuel Graña

Computational Intelligence Group,
University of the Basque Country
ramon.moreno@ehu.es
http://www.ehu.es/ccwintco

Abstract. We propose a hybrid gradient which provides a good behavior on regions with different illumination. It avoids the shadow effects focusing on the detection of regions of the scene with different chromatic properties. It works with image intensity and chromaticity according with its intensity level emulating the Human Vision System (HVS). This gradient is grounded in the Hyperspherical coordinates, therefore it has a general propose and can be applied on RGB images, multi-spectral images or hyperspectral images.

Keywords: Hyperspectral, Multispectral, Hyperspherical Coordinates, Image Gradient.

1 Introduction

Edge detection is a key step in image segmentation process, on it depends the results quality. Edge detectors are widely performed by gradients like Sobel, Prewitt, Canny, watershed and then [1,2,3]. There are also other approaches based on clustering like c-means, statistical analysis methods or sophisticated technics for image segmentation like growing-neural-gas [4,5]. The key difference between them is that, whereas the first ones work on the image domain, the second ones work only within the n-dimensional image space.

Nowadays, cost of hyperspectral cameras are turning cheaper, and it provides a new field of applications. On the other hand, hyperspectral images come taking a strong importance in remote sensing. In both cases when applying segmentation process by different strategies, it is very common to detect the shadows and label them like "shadows" [6,7,8,9].

In conventional digital images, it is very usual to work in color spaces, and change between them depending on the application. There are a lot of them RGB, HSI, HSV, CIE L*a*b, CIE L*u*v and then. That is not the case of hyperspectral images, where color spaces and colorimetrycal concepts are not been developed clearly so far.

In this work we will work with the hyperspherical interpretation of the n-dimensional space. It will help us to separate chromaticity and intensity without changing the image space. Hence, the Hyperspherical Coordinates let us perform

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 404–415, 2012.

the main colorimetrical separation. After that we will propose an like-prewitt gradient based on the image chromaticity, which is independent of the intensity and has better behavior on shines and shadows, solving in this manner an old problem in the segmentation of hyperspectral images.

This paper is outlined as follows, Sec. 2 discuses about the Spherical coordinates and its colorimetrical meaning, where Sub-sec. 2 gives how to transform an image in Euclidean coordinates to Hyperspherical coordinates. Sec. 3 refers to gradients, giving a chromatic gradient on Sub-sec.3.1 and other gradient on Sub-sec.3.1. On Sub-sec.3.2 we present the hybrid gradient. In Sec. 4 we show some experimental results, and we will finish this paper in Sec. 5 with the conclusions.

2 Hyperspherical Coordinates

A n-sphere is a generalization of the surface of an ordinary sphere of an arbitrary dimension. n-Spheres are also named Hyperspheres when dimensionality is bigger then 3. We are interested in the Hyperspherical expression of an hyperdimensional point and its meaning under a colorimetrycal point of view. Let us begin this explanation for a three-dimensional space and after that its extension for n-dimensional spaces.

When working in a three-dimensional space like RGB, it easy to see the spherical expression of a point. Figure 1 shows this equivalence. For a color with (r, g, b) values, we can express it by Spherical coordinates as (θ, ϕ, l) where θ and ϕ are the angular parameters and l the vector magnitude.

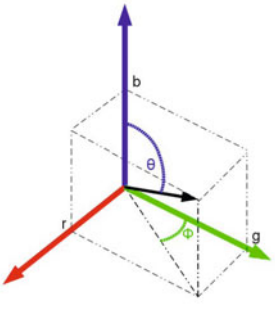


Fig. 1. A vectorial representation of color c in the RGB space

Spherical coordinates provide a good tool when working with color. There are several works where spherical coordinates in three-dimensional color spaces [10,11] estimate the chromaticity of the illumination and detect chromatic edges respectively. For a three-dimensional color space like RGB is very interesting the correspondence between the angular parameters (θ, ϕ) and the chromaticity. In [10] is shown this relationship. In few words, it is summarized as follows: the

angular parameters define an infinite line, and this line is the natural expression of the pixel chromaticity. It means that all points in a line has the same chromaticity. This aspect is important because the Spherical expression of a point within Euclidean space lets us to separate intensity and chromaticity, where l is the intensity and the angular parameters its respective chromaticity.

2.1 From Euclidean to Hyperspherical Coordinates

A pixel p in Euclidean coordinates of n dimensions is expressed by $p = \{v_1, v_2, v_3, ..., v_n\}$ where v_i is the value if the $i-th$ dimension. This pixel can be expressed equivalently by Hyperspherical coordinates as $p = \{l, \phi_1, \phi_2, \phi_3, .., \phi_{n-1}\}$ where l is the longitude vector and $\{\phi_1, \phi_2, \phi_3, .., \phi_{n-1}\}$ are the angular parameters. This transformation is performed uniquely by,

$$
\begin{aligned}
l &= \sqrt{v_1^2 + v_2^2 + v_3^2 + ... + v_n^2} \\
\phi_1 &= \cot^{-1} \frac{v_1}{\sqrt{v_2^2+v_3^2+...+v_n^2}} \\
\phi_2 &= \cot^{-1} \frac{v_2}{\sqrt{v_3^2+v_4^2+...+v_n^2}} \\
&\vdots \\
\phi_{n-2} &= \cot^{-1} \frac{v_{n-2}}{\sqrt{v_{n-1}^2+v_n^2}} \\
\phi_{n-1} &= 2 \cdot \cot^{-1} \frac{\sqrt{v_{n-1}^2+v_n^2}-v_{n-1}}{v_n},
\end{aligned}
$$

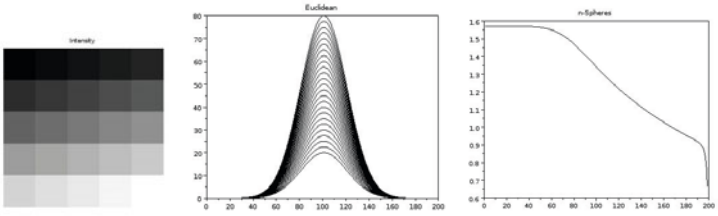
with this exception: if $v_i \neq 0$ for some i but all of $v_{i+1}, \ldots, v_n$ are zero then $\phi_i = 0$.

Let us denote the hiperspherical transformations of a pixel p as $p = \{l, \bar{\phi}\}$ where $\bar{\phi}$ is the vector of size $n-1$ containing the angular parameters. Applying these definitions in a hyperspectral image we can perform the following separation:

Given a hyperspectral image $\mathbf{I}(x) = \left\{(v_1, v_2, v_3, ..., v_n)_x ; x \in \mathbb{N}^2\right\}$, where x refers to the pixel coordinates in the image domain, we denote the corresponding hyperspherical representation as $\mathbf{P}(x) = \left\{(l, \bar{\phi})_x ; x \in \mathbb{N}^2\right\}$, from which we use $\bar{\phi}_x$ as the chromaticity representation of the pixel's and l_x as its respective intensity.

We'll give an example of the previous transformation and its meaning. We have a synthetic hyperspectral image of size 5x5 of domain and 200 bands. Each pixel has the same Gaussian signal but with different intensity. Fig.2(b) shows the spectral signature of all pixels in the Euclidean representation, Fig.2(c) shows the chromatic spectral signature $\bar{\phi}$ of all pixels. We can see only a line because all pixels have the same spectral chromaticity. Finally Fig.2(a) shows the image intensity. If in this image we change each values, the intensity will be different, but the spectral chromaticity of the image will be the same.

Keep in mind an important difference between the Euclidean and Hyperspherical representations. In case of Euclidean representation, a pixel is represented by a "point" in the n-th space. When working in Hyperspherical representation, $\bar{\phi}$ is a "line" the n-th space. l contains the intensity or vector magnitude.



(a) (b) (c)

Fig. 2. Hypersphercal transformation. (a) shows the image intensity, (b) shows a Gaussian signature at different scales, and (c) shows the Hyperspherical transformation of the spectral signatures of all pixels in the image.

Accordingly to the aforegoing transformation, we can perform the following hyperspectral separation. Given a image $\mathbf{I}(x) = \left\{(v_1, v_2, v_3, ..., v_n)_x ; x \in \mathbb{N}^2\right\}$ in the traditional Euclidean representation we can obtain the equivalent image $\mathbf{P}(x) = \left\{(l, \bar{\phi})_x ; x \in \mathbb{N}^2\right\}$ and from $\mathbf{P}(x)$ we can separate $\mathbf{J}(x) = \left\{(l)_x ; x \in \mathbb{N}^2\right\}$ as the image intensity, and $\mathbf{C}(x) = \left\{(\bar{\phi})_x ; x \in \mathbb{N}^2\right\}$ as the image chromaticity. In the synthetic example shown at Fig.2, $\mathbf{I}(x)$ is shown in (b), the spectral chromaticity $\mathbf{C}(x)$ in (c) and the image intensity $\mathbf{J}(x)$ in (a). This separation is very important and its meaning under some models of reflectance like the Dichromatic Reflection Model [12] of the Bidirectional Reflection Distribution Function where they can be decomposed as diffuse and specular components.

3 Gradient

Mathematically, the gradient of elements of a bidimensional space (like images) is given at each image domain point by the derivatives on all directions, usually simplified by the horizontal and vertical directions (in case of images). The gradient function measures the change or variability in a point. The more widely measurement is performed by using intensity, therefore typical gradient expresses the intensity change in a point.

Let us denote $x = (i, j)$ the pixel coordinates. We recall the definition of the image spatial gradient

$$\nabla I(i,j) = \begin{bmatrix} G_i(i,j) \\ G_j(i,j) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial i} I(i,j) \\ \frac{\partial}{\partial j} I(i,j) \end{bmatrix}, \tag{1}$$

where $I(i, j)$ is the image function at pixel (i, j). For edge detection, the usual convention is to examine the gradient magnitude:

$$G(i,j) = |G_i(i,j)| + |G_j(i,j)|. \tag{2}$$

For color images, a simplistic approach to perform edge detection is to drop all color information, and convolve the intensity image with a pair of high-pass

convolution kernels to obtain the gradient components and gradient magnitude. The most popular gradients are the Sobel and the Prewitt. Taking into account spectral information, the straightforward approach is to apply the gradient operators to each band image and to combine the results afterwards $\nabla I = \sum_{i=1}^{n} \nabla I_i / n$.

We disappoint both approaches. With regard to the first one, if we analyze only the intensity image information, we are wasting all spectral information and by other hand it is obvious to understand that there are infinite spectral signatures which can have the same sum intensity, hence two neighbor pixels with different spectral signature but with the same sum intensity are not going to be differentiated. Respect to the second one, is very usual perform gradients which work independently on each band, taking each band as an image intensity, and this approach is not correct because intensity is a property of the full pixel and not a property shared uniformly in the bands. Fig. 3 shows the prewitt gradient following the second approach.

First column shows the original images, where the first one, is a synthetic blue ball over a green background, whereas the second one is a synthetic orange over the same green background. Both images have sun lighting. Second row shows the respective gradients. As we can see, this gradient is absolutely dependent on the intensity, consequently has a strong response on shadows and shines, and a low response on regions poorly illuminated, even on regions with different chromatic properties, like the region borders on the shadows. Last row shows the binary image with the more important edges.

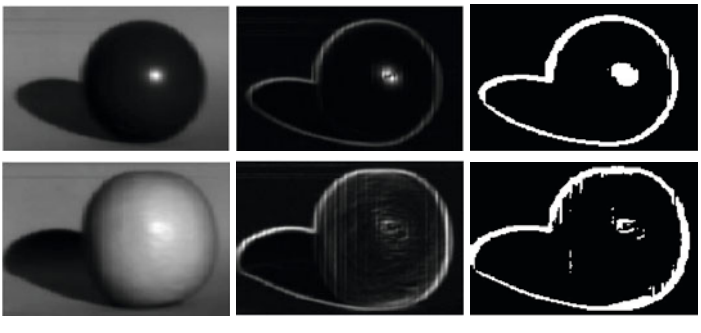


Fig. 3. Classical Prewitt gradient (on each band) of two hyperspectral images. First column shows the grayscale image, second column shows the gradient, and third column shows a binarization of the gradient image.

3.1 Gradient Operators

On digital images, gradients are performed by convolution masks like the masks. Here, the most important is the distance applied between neighbor pixels. e.g. first row of the first matrix of the prewitt mask, means $-a_1 + a_3$, or $a_3 - a_1$ where this difference is calculated by the Euclidean distance. We will define a chromatic

gradient which works with the angles of the Hyperspherical coordinates, and other gradient which works with the Euclidean representation of the image. Let us refer to this second one as e-gradient.

Chromatic Gradient

We propose a chromatic distance for hyperspectral images by using the chromatic image $\mathbf{C}$ previously defined. For two pixels p and q we define the following distance:

$$dc(p,q) = \sqrt{\sum_{i=1}^{n-1} \left\| \bar{\phi}_{p,i} - \bar{\phi}_{q,i} \right\|} \tag{3}$$

where n is the amoung of angles.

This distance is the Manhattan or Taxicab distance of the hypersherical angles of the image. Let us refer it as the chromatic distance between two hyperspectral points.

We will formulate a pair of Prewitt-like gradient pseudo-convolution operations on the basis of the above distance. Note that the $dc(p,q)$ distance is always positive. Note also that the process is non linear, so we can not express it by convolution kernels. The row pseudo-convolution is defined as:

$$CG_R\left(\mathbf{C}\left(i,j\right)\right) = \sum_{r=-1}^{1} dc\left(\mathbf{C}\left(i-r,j+1\right), \mathbf{C}\left(i-r,j-1\right)\right),$$

and the column pseudo-convolution is defined as

$$CG_C\left(\mathbf{C}\left(i,j\right)\right) = \sum_{c=-1}^{1} dc\left(\mathbf{C}\left(i+1,j-c\right), \mathbf{C}\left(i-1,j-c\right)\right),$$

so that the color distance between pixels substitutes the intensity subtraction of the Prewitt linear operator. The color gradient image is computed as:

$$CG(x) = CG_R\left(x\right) + CG_C\left(x\right) \tag{4}$$

Fig. 4 illustrates the chromatic gradient performance. First row shows the orange, Fig. 4(b) is the chromatic gradient and Fig.4(c) is the binarization by Otsu[13] thresholding. As we can see, by difference with Fig.3 regions with different chromatic properties are well detected, despite regions poor illuminated like shadows has gradient response. On the second row we shows the same results over an image of the Forster dataset [14]. Fig. 4(e) is the chromatic gradient and Fig.4(f) is the binarization by Otsu. This gradient offer good results on regions enough illuminated avoiding intensity changes, but by other hand it is unstable on regions without illumination. We will try to solve it taking the best properties of the following gradient.

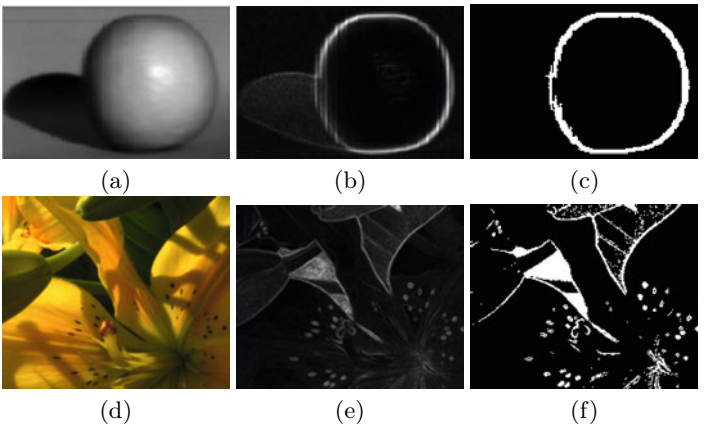


Fig. 4. Chromatic gradient. First column shows the grayscale image, second column shows the chromatic gradient and third column shows a binarization of the gradient image by using Otsu threshold.

Gradient of an Image in Euclidean Coordinates (e-gradient) Analogously to the above gradient, we can define a distance for the original image in the Euclidean coordinates as:

$$id(p,q) = \sqrt{\sum_{i=1}^{n} \|v_{p,i} - v_{q,i}\|} \tag{5}$$

where n is the vector dimensionality.

The row pseudo-convolution is defined as:

$$IG_R\left(\mathbf{I}\left(i,j\right)\right) = \sum_{r=-1}^{1} id\left(\mathbf{I}\left(i-r,j+1\right),\mathbf{I}\left(i-r,j-1\right)\right),$$

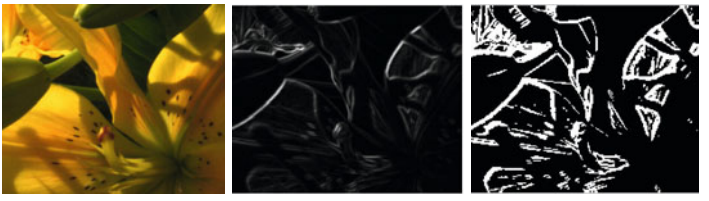
and the column pseudo-convolution is defined as

$$IG_C\left(\mathbf{I}\left(i,j\right)\right) = \sum_{c=-1}^{1} id\left(\mathbf{I}\left(i+1,j-c\right),\mathbf{I}\left(i-1,j-c\right)\right),$$

so that the color distance between pixels substitutes the intensity subtraction of the Prewitt linear operator. The color gradient image is computed as:

$$IG(x) = IG_R\left(x\right) + IG_C\left(x\right) \tag{6}$$

Fig. 5 shows the e-gradient performance. As we can see, it is very sensitive to intensity changes, but it has a good property; by difference with the chromatic gradient, it can detect edges in regions with poor illumination.



(a) (b) (c)

Fig. 5. e-Gradient. (a) Original image, (b) e-Gradient, (c) a binarization of the e-Gradient.

3.2 Hybrid Gradient

Our goal is to develop a hybrid gradient with takes profit of the best properties of each one of the previous gradients. Inspired on the HVS, on the retina, there are two main kinds of cells; cones and rods. The rods are luminance detectors and the cones are chromatic detectors. Both need different energy levels for their activation. Rods need less energy than cones, for this reason human vision becomes grayscale under poor illumination, and colors are better detected with a good illumination.

We'll propose activation function $\alpha(x)$ of the chromatic component. For luminance values below a, the chromatic component of the distance is inactive, for intensity values in the interval $[a, b]$, we smoothly change the contribution of the chromatic component of the hybrid distance from zero to its maximum $c \leq 1$ according to a sinusoidal function. Finally, for intensity values above b its contribution is always c. The three parameters a, b, c are in the range $[0, 1]$.

The function $\alpha(x)$ specifies the mixing of the chromatic and grayscale distances depending on the image intensity. It is defined as follows:

$$\boldsymbol{\alpha}(x) = \begin{cases} 0 & x \leq a \\ \frac{c}{2} + \frac{c}{2}\cos\left(\frac{(x-a)\cdot\pi}{b-a} + \pi\right) & a < x < b \\ c & x \geq b \end{cases} \tag{7}$$

By analogy with HVS we are going to propose the following hybridization function. Fig. 6 shows the scheme. Given a image $\mathbf{I}$ by using the Hyperspherical coordinates we can obtain the chromaticity $\mathbf{C}$ and the intensity $\mathbf{J}$. Applying the Eq.(6) on $\mathbf{I}$ we obtain the image gradient IG. By applying the Eq.(4) on $\mathbf{C}$ we obtain the image gradient CG, and by applying the Eq.(7) on $\mathbf{J}$ we obtain the $\boldsymbol{A}$ matrix which help to perform the hybridization as:

$$HG = \boldsymbol{A} * CG + (1 - \boldsymbol{A}) * IG \tag{8}$$

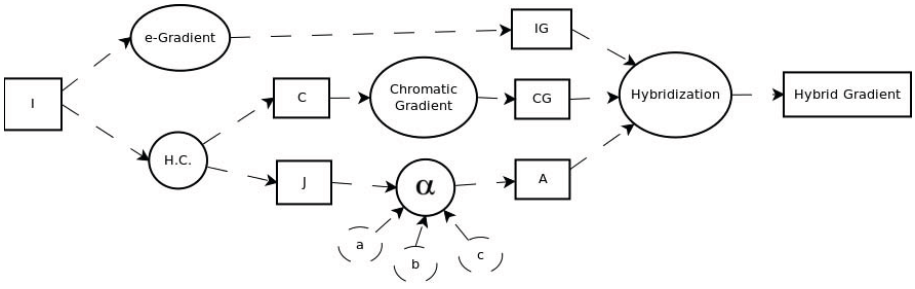


Fig. 6. Scheme of the hybrid gradient

4 Experimental Results

The behavior of the proposed hybrid gradient can be adapted to different images or different expectations of results by using the a, b, c parameters of the α function. For the experiment on this section, we have used the same parameters $a = 0$, $b = 0.1$ and $c = 1$. It means that we are going to perform mainly a chromatic gradient and for regions poorly illuminated we use the e-gradient, as we defend in this work. Nevertheless, by manipulating the α parameters we can obtain different gradients.

We present some experimental results over the images of Foster data set[14]. Fig.7 shows the experimental results. First column contains the original RGB images, second one contains the hybrid gradient output and third row show a binarization based on Otsu thresholding. We can appreciate on the first image, how the digital[1] flower is perfectly detected and differenced from the background, hence it looks homogeneity inside in despite of the shadows of each bell. Worth some words the third image which can differenciate the different chromatic regions with independence of it intensity. If we compare it with the output of the Fig.4 and Fig.5, we can appreciate how the hybrid gradient avoids the shortcoming of the previous gradients and takes profit of the best properties of each ones, and by the hybridization we have a good result; invariance respect to the intensity changes, and edge detection on dark regions. Last image shows a clear example as the shadow effects are avoid by the hybrid gradient. The shadow on the frontage house has not effect on the gradient, whereas edges of dark regions like the houses behind are well detected.

[1] Digital is the name of this flower.

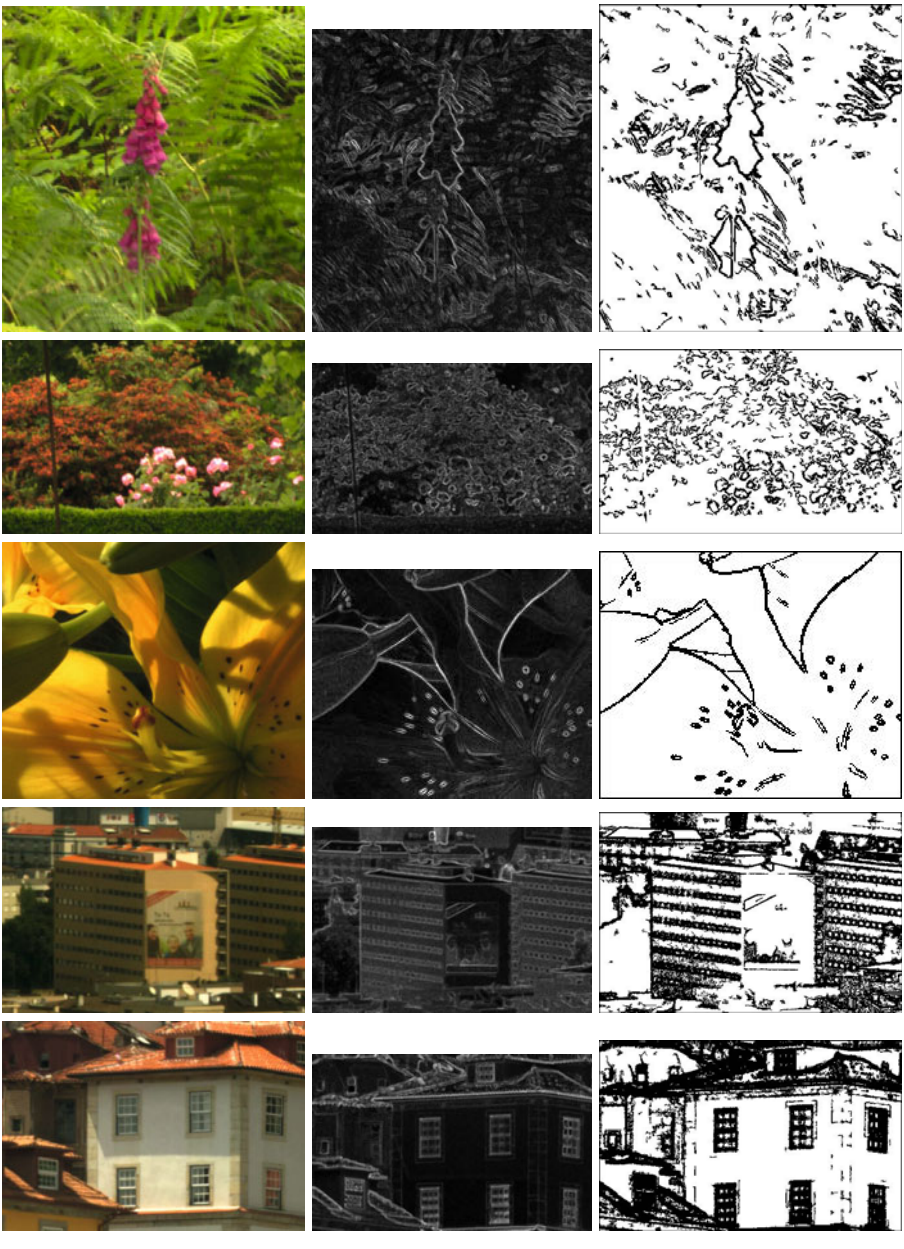


Fig. 7. Hybrid gradient on hyperspectral images. First column shows the grayscale image, second column shows the chromatic gradient and third column shows a binarization of the gradient image based on Otsu threshold.

5 Conclusions

This work presents a hybrid gradient for general purposes. It woks with the image intensity and image chromaticity by two different ways emulating the HVS. It is versatile thanks to the α parameters which can adapt the hybrid gradient performance to different expectations. In addition, the e-gradient can be used on n-dimensional images, and the Hyperspherical coordinates can be applied too on n-dimensional images, therefore, the proposed Hybrid gradient can be used on all images (independent of its amount of bands). In fact on previous works [15] we are applied this ideas on RGB images, the ball images shown on Fig.3 have 128 bands and Foster images have 33 bands.

References

1. Hildreth, E.C.: Edge detection. Technical Report, vol. 207, pp. 187–217. Massachusetts Institute of Technology Cambridge (1985)
2. Wang, D.: A multiscale gradient algorithm for image segmentation using watersheds. Pattern Recognition 30(12), 2043–2052 (1997)
3. Nezhadarya, E., Ward, R.K.: A new scheme for robust gradient vector estimation in color images. IEEE Transactions on Image Processing 20(8), 2211–2220 (2011)
4. Angelopoulou, A., Psarrou, A., Garcia Rodriguez, J., Gupta, G.: Active-gng: model acquisition and tracking in cluttered backgrounds. In: Proceeding of the 1st ACM Workshop on Vision Networks for Behavior Analysis, VNBA 2008, pp. 17–22. ACM, New York (2008)
5. García-Rodríguez, J., García-Chamizo, J.M.: Surveillance and human-computer interaction applications of self-growing models. Applied Soft Computing, 4413–4431 (2011) (in Press, Corrected Proof)
6. Graña, M., Villaverde, I., Maldonado, J.O., Hernandez, C.: Two lattice computing approaches for the unsupervised segmentation of hyperspectral images. Neurocomputing 72(10-12), 2111–2120 (2009)
7. Plaza, A., Benediktsson, J.A., Boardman, J.W., Brazile, J., Bruzzone, L., Camps-Valls, G., Chanussot, J., Fauvel, M., Gamba, P., Gualtieri, A., Marconcini, M., Tilton, J.C., Trianni, G.: Recent advances in techniques for hyperspectral image processing. Remote Sensing of Environment 113(suppl.1), 110–122 (2009)
8. Tuia, D., Kanevski, M., Munoz-Mari, J., Camps-Valls, G.: Structured output SVM for remote sensing image classification. In: IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2009, pp. 1–6. IEEE (2009)
9. Tarabalka, Y., Chanussot, J., Benediktsson, J.A.: Segmentation and classification of hyperspectral images using watershed transformation. Pattern Recogn. 43(7), 2367–2379 (2010)
10. Moreno, R., Graña, M., d'Anjou, A.: Illumination source chromaticity estimation based on spherical coordinates in rgb. Electronics Letters 47(1), 28–30 (2011)
11. Moreno, R., Graña, M., Zulueta, E.: RGB colour gradient following colour constancy preservation. Electronics Letters 46(13), 908–910 (2010)
12. Shafer, S.A.: Using color to separate reflection components. Color Research and Aplications 10, 43–51 (1984)

13. Otsu, N.: A threshold selection method from Gray-Level histograms. IEEE Transactions on Systems, Man and Cybernetics 9(1), 62–66 (1979)
14. Foster, D.H., Nascimento, S.M., Amano, K.: Information limits on neural identification of colored surfaces in natural scenes. Visual neuroscience 21(3), 331–336 (2004) PMID: 15518209 PMCID: 1991287
15. Moreno, R., Graña, M., d'Anjou, A.: A Hybrid Color Distance for Image Segmentation. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS, vol. 6679, pp. 447–454. Springer, Heidelberg (2011)

A Hybrid Segmentation of Abdominal CT Images

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. The weakening of the aortic wall leads to its deformation and the generation of a thrombus. 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. In order to effectively assess the changes experienced after surgery, it is necessary to segment the aneurysm, which is a very time-consuming task. Here we describe the initial results of a novel active learning hybrid approach for the semi-automatic detection and segmentation of the lumen and the thrombus of the AAA, which uses image intensity features and discriminative Random Forest classfiers.

Keywords: Medical Image, Segmentation, Active Learning.

1 Introduction

Generally, an AAA is considered to be present when the minimum anteriposterior diameter of the aorta reaches 3.0 cm[13].

The majority of aortic aneurysms are asymptomatic and without complications. Aneurysms that cause symptoms have a higher risk of rupture. Abdominal pain or back pain and are the two main clinical features suggestive of either the recent expansion or leakage. The complications are often life threatening and can occur in a short space of time. Therefore, the challenge is to diagnose before the onset of symptoms. Asymptomatic aneurysms are often detected incidentally[15].

The prevalence of AAA depends on various risk factors, as advancing age, family history, male gender, and tobacco use. According to the ACC/AHA guidelines (Hirsch et al., 2006) the prevalence of AAA 2.9 to 4.9 cm in diameter ranges from 1.3 % for men aged 45 to 54 years up to 12.5 % for men 75 to 84 years of age. Comparable prevalence figures for women are 0% and 5.2 %, respectively. In these studies the AAA was defined as an aortic diameter $\geq$3cm. If clinically important aneurysms are only taken into account (AAA measuring $\geq$ 4 cm in diameter) the indicated prevalence would be lower.

Even though several segmentation methods for vascular structures have been developed [17] [18], the segmentation of the AAA thrombus is still a challenging task due to the similar intensity values of the aneurysm thrombus and its surrounding tissue (Fig. 1). Several AAA thrombus segmentation methods have been recently developed. The method by de Bruijne et al.[5] is an interactive

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 416–423, 2012.

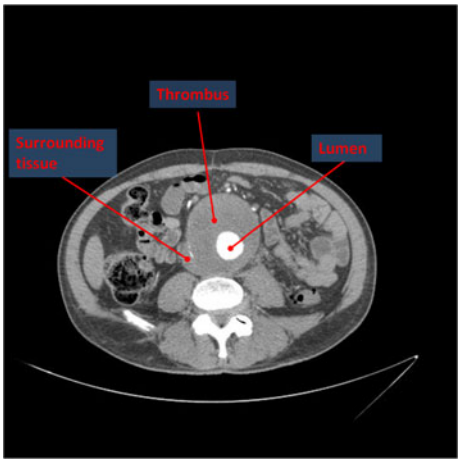


Fig. 1. Axial view of thrombus and lumen in a CT orthoslice using the contrast agents, blood in lumen is highlighted but thrombus intensity levels are similar to other surrounding tissue

contour tracking method for axial slices; Olabarriaga et al.[20] employ a deformable model approach based on a nonparametric statistical grey level appearance model of an prior lumen contour shape segmented interactively; Zhuge et al. [23] present a level-set segmentation based on a parametric statistical model; Demirci et al. [9] propose a deformable B-spline parametric model based on a nonparametric intensity distribution model and; Freiman et al. [11] apply a an iterative model-constrained graph-cut algorithm.

These methods all involve a significant user interaction or initialization, and there are differences between them but in our opinion the methods are more similar than different. All define a dynamics (or an optimization) of auxiliary variables associated with the pixels. Each variable is updated depending on a linear combination of variables from neighboring pixels, as well as some kind of nonlinear operation.

In this paper we propose a machine learning based method. Machine Learning has yielded superior performance in the Berkeley Segmentation Benchmark [19] comparing to other methods. In order to get a small as posible training set, we apply the active learning approach. Then we train the model and perform the classification of the pixels of the entire image with random forest classifier. Several authors have been developing RF based image segmentation techniques in the last years. Lempitsky et al. [16] have used the binary RF to automatically delineate the myocardium in 3D ultrasound (US) of adult hearts. Yi et al. [22]segmented three brain tissues from MRI volumes using the RF technique. Geremia et al. [12] have used the RF technique to segment multiple sclerosis in multi-channel brain MR images. Yaqub et al.[21] proposed a weighted RF technique in which weights are assigned to trees during testing depending on their strength to classify a new test case. On the other hand, Criminisi et al. [7,8]

have used the standard RF technique to automatically detect several organs in CT volumes by finding a 3D bounding box around each organ.

2 Methods

This section describes the active learning procedure and the random classification forest we use for the segmentation process of the AAA.

Our detection and segmentation problem can be described as a multiclass classification of voxel samples into aortic lumen, thrombus, bones (column) and background. We perform the classification with a supervised method: discriminative random decision forest. We build the training data in an iterative active learning process (Fig. 2).

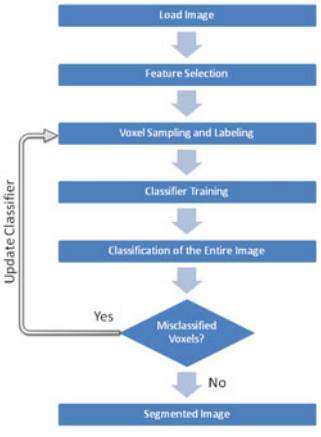


Fig. 2. Pipeline of the active learning segmentation process

2.1 Active Learning

In the current state of the art, the use of statistical learning models is a common practice for other research areas like remote sensing; Support Vector Machines (SVM) [10] or neural networks[6] algorithms are widely used for the classification. However, the performances of supervised algorithms strongly depend on the information gain provided by the data used to train the classifier. This makes the construction of the training set a cumbersome task requiring extensive manual analysis of the image. This is typically done by visual inspection of the scene and successive labeling of each sample. Consequently, the training set is highly redundant and training phase of the model is significantly slowed down. Besides, noisy pixels may interfere the class statistics, which may lead to poor classification performances and/or overfitting. For these reasons, a training set should also be kept as small as possible and focused on those pixels effectively

improving the performance of the model. Therefore a desirable training set must be constructed in a smart way, meaning it must represent correctly the class boundaries by sampling discriminative pixels. In the machine learning literature this approach to sampling is known as active learning[21].

Active learning focuses on the interaction between the user and the classifier. The model returns to the user the pixels whose classification outcome is the most uncertain. After accurate labeling by the user, pixels are included into the training set in order to reinforce the model[12]. This way, the model is optimized on well-chosen difficult examples, maximizing its generalization capabilities.

2.2 Random Forest Classifiers

The random forests (RF) machine learning algorithm is a classifier [4] that encompasses bagging [3] and random decision forests [1] [14]and is widely used in a variety of aplications [2]. RF became popular due to its simplicity of training and tuning while offering a similar performance to boosting. It is a large collection of decorrelated decision trees, which are ideal candidates to capture complex interaction structures in data. RF is supposed to be resistant to over-fitting of data if individual trees are sufficiently deep. Consider a RF collection of tree predictors

$h(x;\psi_u), u = 1, ..., U$, where x is a random sample of d-dimensions associated to random vector X and ψ_u independent identically distributed random vectors. Given a dataset of N samples, the bootstrap training sample of tree $h(x;\psi_u)$ is used to grow the tree by recursively selecting a subset of random dimensions $\hat{d}$ such that $\hat{d} \ll d$ d and picking the best split of each node based on these variables. Unlike conventional decision trees, pruning is not required.

$$\hat{c} = majority\, vote\{C_u(x)_1^u\} \tag{1}$$

To make a prediction for a new sample x, the trained RF could then be used for classification by majority vote among the trees of the RF as shown in Eq. (1), where $C_u(x)$ is the class prediction of the uth RF tree. The important parameters of the RF classifier were set as follows in this case. The number of trees in the forest should be sufficiently large to ensure that each input class receives a number of predictions: set to 200. The number of variables randomly sampled at each branch: set to 5.

2.3 Feature Set Construction through Active Learning

We build a minimal set of manually labeled voxels from the CT images. We use the phrase feature set to denote the group of features of the same type but with different dimensions and locations around a pixel of interest. Initially we choose a wide variety of the most common image intensity features (Table 1) as well as different radius values (1,2,4...2^n) around the voxel of interest.

However the training set build this way has an extend number of attributes, that is convenient to get a less expensive computational task. Since many of

Table 1. Initially chosen Image features to build the training set

Image Feature
Gaussian blur
Sobel filter
Hessian
Difference of gaussians
Mean
Variance
Median
Maximum
Minimum
Laplacian

Table 2. Finally selected Image features to build the training set

Image Feature
Gaussian blur
Mean
Median
Maximum
Laplacian

the possible features provide relatively poor information gain, the probability to select a good feature is low. If we compute the information gain provided by each feature and radius value, and we choose the features that give as more information (Table 2), we get a smaller set that produces similar classification results.

The radius of the filters we select are: 4,8 and 16.

3 Results

Results presented in this section aim at evaluating the segmentation results. Our method has been experimentally tested on a real human contrast-enhanced dataset obtained from a LightSpeed16 CT scanner (GE Medical Systems, Fairfield, CT, USA) with 512x512x354 voxel resolution and 0.725x0.725x0.8 mm spatial resolution. We train over the set of features different classifiers and we show the results for accuracy, area under the ROC (AUC) and the residual minimum square error (RMSE). Ten-fold cross validation is used in every experiment. We perform 10 iterations in order to avoid testing errors. The final result is the average outcome of the 10 iterations on the test data. We obtain the best results for Random Forest followed by Learning Model Tree (LMT), Multi Layer Perceptron (MLP) and Bayes Networks while Radial Basis Function (RBF) and Support Vector Machines (SVM) give us the less accurate results (Table 2).

Table 3. Cross-validation results over the abdominal image features computed from the CT datasets for thrombus segmentation

Classifier	Accuracy	AUC	RMSE
SVM	77.4663	0.830	0.2828
MLP-BP	92.3157	0.968	0.1725
RBF	76.6355	0.853	0.2771
LMT	96.6771	0.976	0.1251
Bayes-Net	83.9045	0.938	0.2644
Random Forest	98.0270	0.999	0.0938

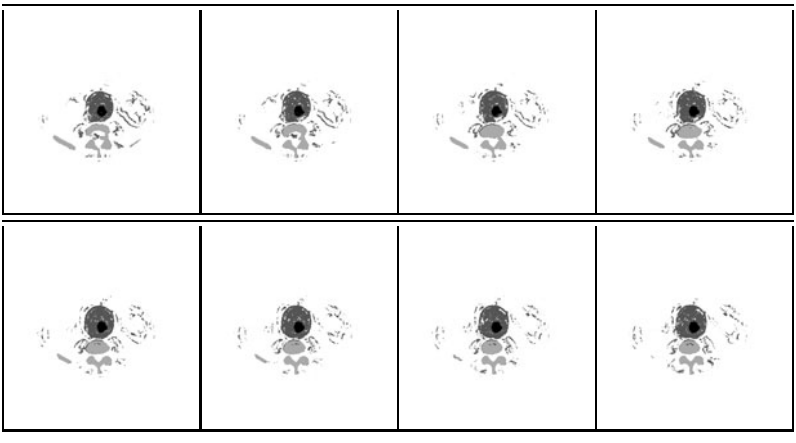


Fig. 3. 8 consecutive abdominal segmented images of the same patient after RF classification process. Lumen (darkest circle in the center), thrombus (circle around lumen), and bones (backbone and rib) are distinguished.

The classifier build with the training set corresponding to the image features of just one slice detects and segments the anatomical structures in several consecutive slices

4 Conclusion and Future Works

We use Active learning techniques to build feature sets. After evaluating the information gain provided by the a variety of intensity based features we choose a set of features to train the classifier and perform the voxel-based segmentation.

We compare the Random Forest classifier with other common classifiers and neural networks and demonstrate that it carry out more accurate results and thus provide a efficient tool for discriminating voxels corresponding to specific anatomical structures in abdominal CT images. Specifically, we discriminate efficiently the voxels corresponding to the thrombus.

Our method is being currently tested on real human datasets and results are promising. Accurate segmentations are obtained in areas where it is difficult to distinguish the thrombus from surrounding structures and is a good input for a generative model that would improve the segmentation quality.

Future work will be oriented to improve the training set, fine-tune the parameters of the process for a large number of datasets and validate the segmentation by comparison with manual segmentation and other methods.

References

1. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural Computation 9(7), 1545–1588 (1997)
2. Barandiaran, I., Paloc, C., Graña, M.: Real-time optical markerless tracking for augmented reality applications. Journal of Real-Time Image Processing 5, 129–138 (2010)
3. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. de Bruijne, M., van Ginneken, B., Viergever, M.A., Niessen, W.J.: Interactive segmentation of abdominal aortic aneurysms in cta images. Med. Image Anal. 8(2), 127–138 (2004)
6. Bruzzone, L., Prieto, D.F.: An incremental-learning neural network for the classification of remote-sensing images. Pattern Recognition Letters 20(11-13), 1241–1248 (1999)
7. Criminisi, A., Shotton, J., Bucciarelli, S.: Decision forests with long-range spatial context for organ localization in ct volumes. In: MICCAI Workshop on Probabilistic Models for Medical Image Analysis (2009)
8. Criminisi, A., Shotton, J., Robertson, D., Konukoglu, E.: Regression forests for efficient anatomy detection and localization in ct studies. In: Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging, pp. 106–117 (2011)
9. Demirci, S., Lejeune, G., Navab, N.: Hybrid deformable model for aneurysm segmentation. In: ISBI 2009, pp. 33–36 (2009)
10. Fauvel, M., Benediktsson, J., Chanussot, J., Sveinsson, J.: Spectral and spatial classification of hyperspectral data using svms and morphological profiles. IEEE Transactions on Geoscience and Remote Sensin 46(11), 3804–3814 (2008)
11. Freiman, M., Esses, S.J., Joskowicz, L., Sosna, J.: An Iterative Model-Constraint Graph-cut Algorithm for Abdominal Aortic Aneurysm Thrombus Segmentation. In: Proc. of the 2010 IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro (ISBI 2010), pp. 672–675. IEEE, Rotterdam (2010)
12. Geremia, E., Menze, B.H., Clatz, O., Konukoglu, E., Criminisi, A., Ayache, N.: Spatial Decision Forests for MS Lesion Segmentation in Multi-Channel MR Images. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010. LNCS, vol. 6361, pp. 111–118. Springer, Heidelberg (2010)
13. Hirsch, A., Haskal, Z., Hertzer, N., Bakal, C., Creager, M., Halperin, J., Hiratzka, L., Murphy, W., Olin, J., Puschett, J., et al.: Acc/aha 2005 practice guidelines for the management of patients with peripheral arterial disease. Circulation 113(11), e463 (2006)
14. Ho, T.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)

15. Karkos, C., Mukhopadhyay, U., Papakostas, I., Ghosh, J., Thomson, G., Hughes, R.: Abdominal aortic aneurysm: the role of clinical examination and opportunistic detection. European Journal of Vascular and Endovascular Surgery 19(3), 299–303 (2000)
16. Lempitsky, V., Verhoek, M., Noble, J., Blake, A.: Random forest classification for automatic delineation of myocardium in real-time 3d echocardiography. Functional Imaging and Modeling of the Heart, 447–456 (2009)
17. Lesage, D., Angelini, E.D., Bloch, I., Funka-Lea, G.: A review of 3d vessel lumen segmentation techniques: Models, features and extraction schemes. Medical Image Analysis 13(6), 819–845 (2009)
18. Macia, I., Grana, M., Maiora, J., Paloc, C., de Blas, M.: Detection of type ii endoleaks in abdominal aortic aneurysms after endovascular repair. Computers in Biology and Medicine (2011)
19. Martin, D.R., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5), 530–549 (2004)
20. Olabarriaga, S., Rouet, J., Fradkin, M., Breeuwer, M., Niessen, W.: Segmentation of thrombus in abdominal aortic aneurysms from CTA with nonparametric statistical grey level appearance modeling. IEEE Transactions On Medical Imaging 24(4), 477–485 (2005)
21. Yaqub, M., Javaid, M., Cooper, C., Noble, J.: Improving the classification accuracy of the classic rf method by intelligent feature selection and weighted voting of trees with application to medical image segmentation. Machine Learning in Medical Imaging, 184–192 (2011)
22. Yi, Z., Criminisi, A., Shotton, J., Blake, A.: Discriminative, Semantic Segmentation of Brain Tissue in MR Images. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) MICCAI 2009. LNCS, vol. 5762, pp. 558–565. Springer, Heidelberg (2009)
23. Zhuge, F., Rubin, G.D., Sun, S.H., Napel, S.: An abdominal aortic aneurysm segmentation method: Level set with region and statistical information. Medical Physics 33(5), 1440–1453 (2006)

Hybrid Computational Methods for Hyperspectral Image Analysis

Miguel A. Veganzones and Manuel Graña

Grupo de Inteligencia Computacional, Universidad del País Vasco, Spain

Abstract. In this paper we provide a brief review of recent advances in computational methods for hyperspectral image analysis with emphasis in hybrid approaches. Hyperspectral imagery acquisition and hyperspectral analysis are growing fields. The analysis of hyperspectral images will have an increasing impact in several application areas, i.e., Earth observation, planetology, food industry, quality processes, medicine, etc. Hyperspectral image analysis is itself a hybrid process that chains different computational techniques. We focus on dimensionality reduction and spectral unmixing which are fundamental parts of hyperspectral image analysis.

Keywords: Remote sensing, hyperspectral, computational methods, unmixing, endmember induction.

1 Introduction

Spectral imaging [1] refers to the collection of optical images taken in multiple wavelength bands that are spatially aligned such that at each pixel there is a vector representing the response to the same spatial location for all wavelengths. Hyperspectral imaging systems (HSI) differ from color and multispectral imaging systems (MSI) in the number of bands, color and MSI images have three to ten spectral bands while HSI images tend to have hundred of co-registered bands, spectral resolution, color and MSI system's spectral resolution is on the order of 10 while HSI systems on the order of 100, and contiguty, MSI systems have their spectral bands widely and irregularly spaced while HSI systems have contiguous and regularly spaced bands. While color and MSI systems analysis techniques are very related to the spatial characteristics of the data and they usually deal with each spectral band individually, techniques used with hyperspectral imagery exploit the spectral information contained in the hundreds of contiguous and regularly spaced bands that can be seen as a continuous spectrum measured for each pixel.

Hyperspectral image analysis is a growing field of interest with an increasing impact in different areas [2]. Earth observation is the field in which hyperspectral image analysis has had a major impact by its capacity to exploit patterns of emission and absorption of different materials such as vegetation, minerals, atmospheric components, and so on. Furthermore, the impact of hyperspectral

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 424–435, 2012.

image analysis potentiality has extended to other fields: in planetology hyperspectral data has proven to be of high interest for Mars researches; in industrial quality processes, such as food industry, where hyperspectral information can be of great value; medicine, biochemistry, etc.

In this paper we review more than seventy recent works in hyperspectral image analysis published in peer-review international journals. In section 2 we define the hyperspectral image analysis as a hybrid process where dimensionality reduction and spectral unmixing have a main role. In section 3 we make a survey of novel dimensionality reduction methods. Finally, in section 4 we review methods for spectral unmixing.

2 The Hyperspectral Image Analysis as a Hybrid Process

An hyperspectral image analysis process can often be described as a chain of different computational techniques. Figure 1 shows a flow chart of a common hyperspectral image analysis process. Before hyperspectral image analysis a preprocessing step is usually required. This preprocessing includes hyperspectral sensor calibration and image acquisition, as well as data geo-registration and atmospheric correction if the sensor is mounted in a platform for Earth Observation. The acquired data must be indexed and stored together to some metadata containing information about the sensed scene. Hyperspectral analysis is done over the hyperspectral image and we suppose no any other preprocessing is required.

Dimensionality reduction (DR) is usually a first step in the analysis given the high spectral dimensionality of hyperspectral data. In DR we try to find a low-dimensional optimal representation of the hyperspectral data that could be easily analyzed. *Spectral unmixing* (SU) is a common technique in hyperspectral analysis. In SU the hyperspectral image can be seen as a (non-)linear mixture of the spectral signatures of the materials in the image (endmembers) and their fractional spatial distributions (abundances) plus additive noise. Given the endmembers presented in an hyperspectral image the unmixing looks for the fractional abundances of such endmembers. The abundances obtained by the SU can be the goal of the hyperspectral analysis or an intermediate step to further subpixel analysis. SU requires to know the endmembers of the materials in the image, information that is rarely known a-priori. Thus, SU is often a hybrid process that requires the estimation of the endmembers in the image by means of manual or automatic methods to make the unmixing process possible. Machine learning techniques, such as classification, clustering, target detection and so on, can be applied directly to the hyperspectral data or to the results of DR and/or SU methods. Specific methodologies have been proposed on the literature to deal with the inherent issues of hyperspectral data, and once again, some of them include the combination of different computational techniques as we'll show in next sections.

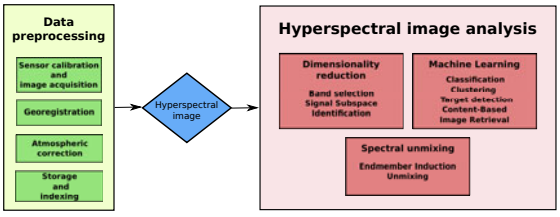


Fig. 1. Diagram of an hyperspectral image analysis

3 Dimensionality Reduction

Hyperspectral imaging implies measuring a large, hundreds or thousands, of spectral bands. Hyperspectral high-dimensional data incorporate redundancies due to the high between bands correlation. The actual spectral information lies in a lower dimensional subspace. Identifying such a subspace, a task called *Signal Subspace Identification* (SSI), is the main way to perform dimensionality reduction on hyperspectral data. However, finding the correct subspace dimensionality in which the hyperspectral data are localized is a very challenging problem. *Dimensionality Reduction* (DR) is often the first step in hyperspectral analysis. DR allows to have a better, more compact, representation of the data alleviating the computational cost and the memory allocation requirements of posterior analysis methods.

SSI can be assessed in two ways: space transformation and band selection. The former implies a transformation of the high-dimensional data space onto a different space whose axes are ordered given some criteria. The dimensionality reduction is achieved by selecting the first N_D axes. The later consists in a selection of a reduced number of bands, N_B, from the original set of bands. The selected spectral bands must contain most of the relevant information given some criteria. In both cases, the identified subspace has a dimensionality, N_D or N_B, lower than the original spectral space.

3.1 SSI by Space Transformation

The most popular SSI by space transformation method is Principal Component Analysis (PCA). PCA calculates the eigenvectors and eigenvalues of the sample data covariance matrix. DR is achieved by selecting those eigenvectors with largest eigenvalues, that is, selecting those eigenvectors (subspace axes) that maximize the variance of the data projected onto them. Maximum Noise Fractions (MNF) and Singular Value Decomposition (SVD) are techniques similar to PCA, that also estimate second-order statistics, but respectively maximizing Signal-to-Noise Ratio (SNR) and maximum power instead of variance. PCA, MNF and SVD have proven not to be suitable for hyperspectral data [3,4]. Hyperspectral images contain many subtle materials with subpixel sizes that are missed by second-order statistics. Recent techniques try to address this second-order-statistics hyperspectral DR problems like rare signal preservation. As it

has been mentioned before rare signals are ignored by traditional second-order statistics methods. Some novel DR methods [5,6,7] try to preserve rare signals combining second-order-statistics with techniques based on the l_2^∞ norm, dividing the subspace in two: the signal subspace and the rare vector subspace.

Independent Component Analysis (ICA) is another well-known technique that have been proposed for hyperspectral DR [8]. ICA looks for statistical independent sources using high-order statistics or mutual information-based criteria. However, statistical independence is a questionable supposition in hyperspectral data [9]. Progressive dimensionality reduction by transform (PRDT) [10] performs DR in terms of progressive information preservation. Each spectral transformed component is ranked respect to its information preservation capability. Two procedures are proposed to perform DR. One starts by a reduced number of spectral transformed components and expands it progressively. The other starts by a large number of components and progressively removes them.

Some recent papers introduce techniques in Remote Sensing field like manifolds and tensors. Works in [11,12] propose improvements to the ISOMAP method [13] to find the non-linear structure of hyperspectral data, that is, a manifold coordinate system that preserves geodesic distances in the high-dimensional hyperspectral data-space. [14] proposes the use of tensors to jointly process spatial and spectral information for denoising and DR.

Recent techniques make use of a-priori information about the data, in the form of partial labeling, to perform SSI. In [15] SSI is achieved by looking for the subspace that minimizes two terms: a discriminative term that assesses the pairwise class separability of the labeled data samples, and a regularization term that characterizes some property of the original data space. The paper proposes two approaches for the regularization term, maximizing data global variance and minimizing reconstruction errors. In [16], authors pursue both DR and classification in a row by pruning a neural network input layer. [17] proposes project labeled samples onto a new 'prototype space' defined by the bands and cluster projected data in order to perform DR by grouping bands containing similar information. [18] proposes improvements to two extensions of Fisher's linear discriminant analysis (LDA), the non-parametric discriminant analysis and the non-parametric weighted feature extraction, to reduce the dimensionality and increase the classification accuracy. The use of generalized relevance learning vector quantization (GRLVQ) to extract those features interesting to classification purposes is proposed in [19].

3.2 SSI by Band Selection

The common feature selection sub-optimal search strategies, like sequential forward selection, steepest ascent and fast constrained search, present the same problems than common subspace identification algorithms. [20] models the hyperspectral DR as a band selection problem where subsets of the original bands are selected and averaged to form a spectral band. The paper proposes modifications of the common feature selection strategies to find the optimal solution minimizing the probability of classification error in the spectral space.

Some methods take advantage of labeled data samples. [21] makes use of a well-know algorithm for target detection, the constrained energy minimization (CEM) algorithm, to linearly constraint a band image while minimizing band correlation or dependence provided by other band images. In [22] authors evaluate labeled sample data statistical dependence by the so-called Hilbert-Schmidt independence criterion (HSIC). The labeled samples are used to calculate the Hilbert-Schmidt norm of the cross-covariance kernel operator. Then, the proposed method looks for those bands that minimize the associated HSIC p-value.

Other recent works make use of pattern of known spectral signatures that are of interest for the posterior analysis. [23] views the hyperspectral sensing process as a projection of the scene space, the scene of all spectral patterns of interest, onto the spectral bands, called the sensor space. Authors provide a method based on the canonical correlation feature selection (CCFS) algorithm to find optimal superpositions of the spectral bands representing the most informative directions in the sensor space for specific patterns in the presence of noise. [24] only uses the spectral band-to-band correlation within a single spectral signature and proposes a method based on orthogonal subspace projections (OSP) to select a variable number of different bands for each of the spectral signatures of interest.

In [25] authors follow a different approach. They modify the sparsity promoting iterated constrained endmember (SPICE) algorithm to perform band selection, endmember induction and spectral unmixing simultaneously. Band selection is achieved by incorporating band weights and a band sparsity promoting term to the SPICE objective function.

4 Spectral Unmixing

The hyperspectral imagery could be seen as a linear mixing model where an hyperspectral image is the result of the linear combination of the pure spectral signature of ground components, named endmembers, with abundance matrices. Let $E = [\mathbf{e}_1, \ldots, \mathbf{e}_p]$ be the endmember signatures where each $\mathbf{e}_i \in R^d$ is a d-dimensional vector. Then, the hyperspectral signature $\mathbf{h}(x, y)$ at the pixel with coordinates (x, y) is defined by the expression:

$$\mathbf{h}(x, y) = \sum_{i=1}^{p} \mathbf{e}_i a_i(x, y) + \eta \tag{1}$$

where $\mathbf{a}(x, y)$ is the n-dimensional vector of fractional abundance at pixel (x, y) and η is the independent additive noise component. There are two common constraints to equation 1: the abundance non-negative constraint (ANC) and the abundance sum-to-one constraint (ASC), respectively defined as $a_i(x, y) \geq 0$, for all $1 \leq i \leq p$, and $\sum_{i=1}^{p} a_i = 1$. Given the endmembers, spectral unmixing looks for the fractional abundances of such endmembers. In order to perform spectral unmixing, the spectral signatures of the materials in the image must be known. This is not the common scenario and often some method must be used to select endmembers from an spectral library or automatically induce them from the image itself.

4.1 Endmember Induction

The linear mixing model can be geometrically interpreted as a simplex in the spectral space whose vertexes are defined by the endmembers in the image. There are a lot of endmember induction algorithms (EIAs) in the literature defining different criteria and methods to find such a simplex. Recent works present new geometrical simplex-based methods or improvements to the existing ones: [26] proposes a minimum volume enclosing simplex (MVES) algorithm for highly mixed data and [27] extends it by a robust MVES (RMVES) that deals with uniform/nonuniform additive Gaussian noise, [28] proposes a simplex growing algorithm (SGA) that works in real time and [29] a SGA based on the Householder transformation, [30] employs support vector machines to cluster the data and build simplexes enclosing each cluster, [31] proposes a simplex-based algorithm that improves endmember induction in the presence of anomalous materials, [32,33] are versions of the N-FINDR algorithm [34]. Lattice computing [35,36,37,38,39] is an alternative to geometrical simplex formulation, where a connection between linear mixing model algebraic properties and lattice independence is established. Non-negative matrix factorization (NMF) is another alternative technique that have attracted recently a lot of attention [40,41,42,43] and that exploits the positive matrix representation of hyperspectral linear mixing model. Spatial information is exploited in [44,45,46] to improve endmember induction. Virtual dimensionality (VD) is an utilitarian definition of the rank of the subspace containing the data, and thus defines the number of endmembers in the image. [47,48] are novel techniques to calculate the VD of an hyperspectral image, although VD has been questioned as a good interpretation for the estimate of the number of materials in the image [49].

4.2 Linear and Non-linear Spectral Unmixing

The unmixing process estimates the fractional abundance of the endmembers in an image. Least squares and orthogonal subspace projections are the most used linear unmixing methods. Often, spectral unmixing is performed assuming ANC and ASC constraints, the so-called full-constrained linear spectral unmixing (FCLSU). However, sometimes the unmixing problem is relaxed by dropping one of the constraints or both, called partially-constrained and unconstrained LSU respectively. These methods have some limitations derived from the linearity of the assumed model and deficiencies in the selection of endmembers. [50] proposes a new approach called spectral-angle measure-based spectral unmixing (SAMSU) that uses spectral angle measures to reduce the spectral unmixing error due to spectral within-class confusion derived from the variability on the spectral signatures amplitude. Same variability problem is addressed in [51] by a method based on the Fisher's discriminant null space (FDNS). [52] proposes dividing the hyperspectral scene in small tiles, inducing a set of endmembers for each local tile. Then, all locally induced endmembers are clustered together to find groups before unmixing the image globally. [53] adopts a Bayesian formulation to exploit spatial correlations between pixels. In [54] authors propose an

analytical solution to FCLSU by projecting image pixels onto the simplex formed by the endmembers and [55] proposes the use. of fuzzy membership functions that are equivalent to the least square solution of the FCLSU problem.

Non-linear spectral unmixing (NLSU) has attracted increasing attention in the last years. Kernelization of LSU methods [56] allows to estimate non-linear abundances. The generalized bilinear model (GBM) [57] is a model for non-linear unmixing of hyperspectral data due to multipath effects. [58] studies a Bayesian algorithm to estimate the abundance coefficients and the noise variance of the GBM.[59] presents a simple but effective process to nonlinear unmixing, where the multiplication of each pair of endmembers results in a virtual endmember representing multiple scattering effect during pixel construction process. Virtual endmembers resulting of non-linear relationships between actual endmembers can yield poor unmixing results as it is shown in [60]. [61] proposes a non-linear unmixing method based on relative distances through networks. The assumption is that for a pixel, the abundance fraction of an individual endmember is inversely proportional to its distance to the endmember and proportional to the sum of distances of the other endmembers. In [62] labeled data samples are used to estimate the non-linear abundances of given endmembers through Gaussian synapse artificial neural networks. [63] presents an algorithm capable of extracting endmembers and determining their abundances in hyperspectral imagery under nonlinear mixing assumptions. The algorithm is based upon simplex volume maximization, and uses shortest-path distances in a nearest-neighbor graph in spectral space, hereby respecting the nontrivial geometry of the data manifold in the case of nonlinearly mixed pixels.

4.3 Hybrid Approaches

Some recent methods follow a hybrid approach where endmember induction and spectral unmixing are used together. In [64,65,66] authors try to solve endmember induction and spectral unmixing at the same time by proposing a Bayesian formulation of the problem. [67,68] combine well-know EIAs, the Pixel Purity Index (PPI) and the N-FINDR algorithms respectively, with linear unmixing to optimize the number of induced endmembers. Similar approaches are proposed in [69,70] where genetic algorithms are used to select the optimal number of the set of induced endmembers. In [71] the addressed problem is the induction of endmembers preserving rare materials. In [72] endmember induction and linear unmixing techniques are combined to address content-based image retrieval for large hyperspectral databases. In [73] the spectral unmixing is modeled as a dependent component analysis, and a method based on maximum non-Gaussianity and Parzen windows technique is proposed to separate the dependent sources and estimate their fractional abundances. [74,75] combine a Bayesian self-organizing map (BSOM), a supervised method to induce the endmembers, with a Gaussian mixture model (GMM) in the former and a fuzzy membership (FM) in the later to perform the unmixing. [76,77] combine endmember induction and linear unmixing into a combinatorial optimization problem solve by either particle swarm optimization (PSO) or ant colony optimization (ACO), where unmixing error is

the objective function and the particles/ants represent different sets of pixels for endmember determination. [78] allows spectral unmixing to be performed directly on compressed data without any need to reconstruct hyperspectral imagery prior to analysis.

5 Conclusions

Hyperspectral image analysis (HSI) is a major avenue of research with a big impact in remote sensing and other areas of application. We have surveyed more than seventy recent peer-reviewed international journal publications showing the increasing scientific community interest on the topic. We also have shown that HSI is inherently a hybrid process that combines different techniques such as dimensionality reduction, endmember induction, unmixing, and so on.

References

1. Chang, C.I.: Hyperspectral Data Exploitation: Theory and Applications. Wiley Interscience (2007)
2. Plaza, A., Benediktsson, J.A., Boardman, J.W., Brazile, J., Bruzzone, L., Camps-Valls, G., Chanussot, J., Fauvel, M., Gamba, P., Gualtieri, A., Marconcini, M., Tilton, J.C., Trianni, G.: Recent advances in techniques for hyperspectral image processing. Remote Sensing of Environment 113(suppl.1), 110–122 (2009)
3. Prasad, S., Bruce, L.M.: Limitations of principal components analysis for hyperspectral target recognition. IEEE Geoscience and Remote Sensing Letters 5(4), 625–629 (2008)
4. Bajorski, P.: On the reliability of PCA for complex hyperspectral data. In: Proceedings of WHISPERS, Grenoble, pp. 1–5 (2009)
5. Kuybeda, O., Malah, D., Barzohar, M.: Rank estimation and redundancy reduction of High-Dimensional noisy signals with preservation of rare vectors. IEEE Transactions on Signal Processing 55(12), 5579–5592 (2007)
6. Acito, N., Diani, M., Corsini, G.: A new algorithm for robust estimation of the signal subspace in hyperspectral images in the presence of rare signal components. IEEE Transactions on Geoscience and Remote Sensing 47(11), 3844–3856 (2009)
7. Acito, N., Diani, M., Corsini, G.: Hyperspectral signal subspace identification in the presence of rare signal components. IEEE Transactions on Geoscience and Remote Sensing 48(4), 1940–1954 (2010)
8. Wang, J., Chang, C.: Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. IEEE Transactions on Geoscience and Remote Sensing 44(6), 1586–1600 (2006)
9. Nascimento, J.M., Dias, J.M.: Does independent component analysis play a role in unmixing hyperspectral data? IEEE Transactions on Geoscience and Remote Sensing 43(1), 175–187 (2005)
10. Chang, C., Safavi, H.: Progressive dimensionality reduction by transform for hyperspectral imagery. Pattern Recognition 44(10-11), 2760–2773 (2011)
11. Bachmann, C.M., Ainsworth, T.L., Fusina, R.A.: Exploiting manifold geometry in hyperspectral imagery. IEEE Transactions on Geoscience and Remote Sensing 43(3), 441–454 (2005)

12. Bachmann, C.M., Ainsworth, T.L., Fusina, R.A.: Improved manifold coordinate representations of Large-Scale hyperspectral scenes. IEEE Transactions on Geoscience and Remote Sensing 44(10), 2786–2803 (2006)
13. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
14. Renard, N., Bourennane, S., Blanc-Talon, J.: Denoising and dimensionality reduction using multilinear tools for hyperspectral images. IEEE Geoscience and Remote Sensing Letters 5(2), 138–142 (2008)
15. Chen, S., Zhang, D.: Semisupervised dimensionality reduction with pairwise constraints for hyperspectral image classification. IEEE Geoscience and Remote Sensing Letters 8(2), 369–373 (2011)
16. Zeng, H., Trussell, H.: Constrained dimensionality reduction using a Mixed-Norm penalty function with neural networks. IEEE Transactions on Knowledge and Data Engineering 22(3), 365–380 (2010)
17. Mojaradi, B., Abrishami-Moghaddam, H., Zoej, M.J., Duin, R.P.: Dimensionality reduction of hyperspectral data via spectral feature extraction. IEEE Transactions on Geoscience and Remote Sensing 47(7), 2091–2105 (2009)
18. Huang, H., Kuo, B.: Double nearest proportion feature extraction for Hyperspectral-Image classification. IEEE Transactions on Geoscience and Remote Sensing 48(11), 4034–4046 (2010)
19. Mendenhall, M., Merenyi, E.: Relevance-Based feature extraction for hyperspectral images. IEEE Transactions on Neural Networks 19(4), 658–672 (2008)
20. Serpico, S.B., Moser, G.: Extraction of spectral channels from hyperspectral images for classification purposes. IEEE Transactions on Geoscience and Remote Sensing 45(2), 484–495 (2007)
21. Chang, C.I., Wang, S.: Constrained band selection for hyperspectral imagery. IEEE Transactions on Geoscience and Remote Sensing 44(6), 1575–1585 (2006)
22. Camps-Valls, G., Mooij, J., Scholkopf, B.: Remote sensing feature selection by kernel dependence measures. IEEE Geoscience and Remote Sensing Letters 7(3), 587–591 (2010)
23. Paskaleva, B., Hayat, M.M., Wang, Z., Tyo, J.S., Krishna, S.: Canonical correlation feature selection for sensors with overlapping bands: Theory and application. IEEE Transactions on Geoscience and Remote Sensing 46(10), 3346–3358 (2008)
24. Wang, S., Chang, C.: Variable-Number Variable-Band selection for feature characterization in hyperspectral signatures. IEEE Transactions on Geoscience and Remote Sensing 45(9), 2979–2992 (2007)
25. Zare, A., Gader, P.: Hyperspectral band selection and endmember detection using sparsity promoting priors. IEEE Geoscience and Remote Sensing Letters 5(2), 256–260 (2008)
26. Chan, T., Chi, C., Huang, Y., Ma, W.: A convex Analysis-Based Minimum-Volume enclosing simplex algorithm for hyperspectral unmixing. IEEE Transactions on Signal Processing 57(11), 4418–4432 (2009)
27. Ambikapathi, A., Chan, T.H., Ma, W.K., Chi, C.Y.: Chance-Constrained robust Minimum-Volume enclosing simplex algorithm for hyperspectral unmixing. IEEE Transactions on Geoscience and Remote Sensing (99), 1–16
28. Chang, C., Wu, C., Lo, C., Chang, M.: Real-Time simplex growing algorithms for hyperspectral endmember extraction. IEEE Transactions on Geoscience and Remote Sensing 48(4), 1834–1850 (2010)
29. Liu, J., Zhang, J.: A new maximum simplex volume method based on householder transformation for endmember extraction. IEEE Transactions on Geoscience and Remote Sensing (in press)

30. Filippi, A.M., Archibald, R.: Support vector Machine-Based endmember extraction. IEEE Transactions on Geoscience and Remote Sensing 47(3), 771–791 (2009)
31. Mei, S., He, M., Zhang, Y., Wang, Z., Feng, D.: Improving Spatial-Spectral endmember extraction in the presence of anomalous ground objects. IEEE Transactions on Geoscience and Remote Sensing 49(11), 4210–4222 (2011)
32. Tao, X., Wang, B., Zhang, L.: Orthogonal bases approach for the decomposition of mixed pixels in hyperspectral imagery. IEEE Geoscience and Remote Sensing Letters 6(2), 219–223 (2009)
33. Xiong, W., Chang, C., Wu, C., Kalpakis, K., Chen, H.M.: Fast algorithms to implement N-FINDR for hyperspectral endmember extraction. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 4(3), 545–564 (2011)
34. Winter, M.E., Descour, M.R., Shen, S.S.: N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data, vol. 3753, pp. 266–275. SPIE, Denver (1999)
35. Grana, M., Villaverde, I., Maldonado, J.O., Hernandez, C.: Two lattice computing approaches for the unsupervised segmentation of hyperspectral images. Neurocomput. 72(10-12), 2111–2120 (2009)
36. Ritter, G.X., Urcid, G., Schmalz, M.S.: Autonomous single-pass endmember approximation using lattice auto-associative memories. Neurocomput. 72(10-12), 2101–2110 (2009)
37. Grana, M., Savio, A.M., Garcia-Sebastian, M., Fernandez, E.: A lattice computing approach for on-line fMRI analysis. Image and Vision Computing 28(7), 1155–1161 (2010)
38. Grana, M., Chyzhyk, D., Garcia-Sebastian, M., Hernandez, C.: Lattice independent component analysis for functional magnetic resonance imaging. Information Sciences 181(10), 1910–1928 (2011)
39. Ritter, G.X., Urcid, G.: A lattice matrix method for hyperspectral image unmixing. Information Sciences 181(10), 1787–1803 (2011)
40. Jia, S., Qian, Y.: Constrained nonnegative matrix factorization for hyperspectral unmixing. IEEE Transactions on Geoscience and Remote Sensing 47(1), 161–173 (2009)
41. Huck, A., Guillaume, M., Blanc-Talon, J.: Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing 48(6), 2590–2602 (2010)
42. Yang, Z., Zhou, G., Xie, S., Ding, S., Yang, J., Zhang, J.: Blind spectral unmixing based on sparse nonnegative matrix factorization. IEEE Transactions on Image Processing 20(4), 1112–1125 (2011)
43. Yokoya, N., Yairi, T., Iwasaki, A.: Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion. IEEE Transactions on Geoscience and Remote Sensing (in press)
44. Zortea, M., Plaza, A.: Spatial preprocessing for endmember extraction. IEEE Transactions on Geoscience and Remote Sensing 47(8), 2679–2693 (2009)
45. Mei, S., He, M., Wang, Z., Feng, D.: Spatial purity based endmember extraction for spectral mixture analysis. IEEE Transactions on Geoscience and Remote Sensing 48(9), 3434–3445 (2010)
46. Martin, G., Plaza, A.: Region-Based spatial preprocessing for endmember extraction and spectral unmixing. IEEE Geoscience and Remote Sensing Letters 8(4), 745–749 (2011)

47. Chang, C., Xiong, W., Liu, W., Chang, M., Wu, C., Chen, C.C.: Linear spectral mixture analysis based approaches to estimation of virtual dimensionality in hyperspectral imagery. IEEE Transactions on Geoscience and Remote Sensing 48(11), 3960–3979 (2010)
48. Eches, O., Dobigeon, N., Tourneret, J.Y.: Estimating the number of endmembers in hyperspectral images using the normal compositional model and a hierarchical bayesian algorithm. IEEE Journal of Selected Topics in Signal Processing 4(3), 582–591 (2010)
49. Bajorski, P.: Does virtual dimensionality work in hyperspectral images? In: Proceedings of SPIE, Orlando, FL, USA, 73341J–11 (2009)
50. Ben Rabah, Z., Farah, I.R., Mercier, G., Solaiman, B.: A new method to change illumination effect reduction based on spectral angle constraint for hyperspectral image unmixing. IEEE Geoscience and Remote Sensing Letters 8(6), 1110–1114 (2011)
51. Jin, J., Wang, B., Zhang, L.: A novel approach based on fisher discriminant null space for decomposition of mixed pixels in hyperspectral imagery. IEEE Geoscience and Remote Sensing Letters 7(4), 699–703 (2010)
52. Canham, K., Schlamm, A., Ziemann, A., Basener, B., Messinger, D.: Spatially adaptive hyperspectral unmixing. IEEE Transactions on Geoscience and Remote Sensing 49(11), 4248–4262 (2011)
53. Eches, O., Dobigeon, N., Tourneret, J.Y.: Enhancing hyperspectral image unmixing with spatial correlations. IEEE Transactions on Geoscience and Remote Sensing 49(11), 4239–4247 (2011)
54. Heylen, R., Burazerovic, D., Scheunders, P.: Fully constrained least squares spectral unmixing by simplex projection. IEEE Transactions on Geoscience and Remote Sensing 49(11), 4112–4122 (2011)
55. Silvan-Cardenas, J.L., Wang, L.: Fully constrained linear spectral unmixing: Analytic solution using fuzzy sets. IEEE Transactions on Geoscience and Remote Sensing 48(11), 3992–4002 (2010)
56. Liu, K.H., Wong, E., Du, E.Y., Chen, C.C.C., Chang, C.I.: Kernel-Based linear spectral mixture analysis. IEEE Geoscience and Remote Sensing Letters (in press)
57. Fan, W., Hu, B., Miller, J., Li, M.: Comparative study between a new nonlinear model and common linear model for analysing laboratory simulated forest hyperspectral data. International Journal of Remote Sensing 30, 2951–2962 (2009)
58. Halimi, A., Altmann, Y., Dobigeon, N., Tourneret, J.Y.: Nonlinear unmixing of hyperspectral images using a generalized bilinear model. IEEE Transactions on Geoscience and Remote Sensing 49(11), 4153–4162 (2011)
59. Raksuntorn, N., Du, Q.: Nonlinear spectral mixture analysis for hyperspectral imagery in an unknown environment. IEEE Geoscience and Remote Sensing Letters 7(4), 836–840 (2010)
60. Chen, X., Chen, J., Jia, X., Somers, B., Wu, J., Coppin, P.: A quantitative analysis of virtual endmembers' increased impact on the collinearity effect in spectral unmixing. IEEE Transactions on Geoscience and Remote Sensing 49(8), 2945–2956 (2011)
61. Karathanassi, V., Sykas, D., Topouzelis, K.N.: Development of a Network-Based method for unmixing of hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing (in press)
62. Pena, F., Crespo, J., Duro, R.: Unmixing Low-Ratio endmembers in hyperspectral images through gaussian synapse ANNs. IEEE Transactions on Instrumentation and Measurement 59(7), 1834–1840 (2010)

63. Heylen, R., Burazerovic, D., Scheunders, P.: Non-Linear spectral unmixing by geodesic simplex volume maximization. IEEE Journal of Selected Topics in Signal Processing 5(3), 534–542 (2011)
64. Dobigeon, N., Tourneret, J.Y., Chang, C.: Semi-Supervised linear spectral unmixing using a hierarchical bayesian model for hyperspectral imagery. IEEE Transactions on Signal Processing 56(7), 2684–2695 (2008)
65. Dobigeon, N., Moussaoui, S., Coulon, M., Tourneret, J., Hero, A.: Joint bayesian endmember extraction and linear unmixing for hyperspectral imagery. IEEE Transactions on Signal Processing 57(11), 4355–4368 (2009)
66. Eches, O., Dobigeon, N., Mailhes, C., Tourneret, J.: Bayesian estimation of linear mixtures using the normal compositional model. application to hyperspectral imagery. IEEE Transactions on Image Processing 19(6), 1403–1413 (2010)
67. Chang, C., Wu, C., Chen, H.: Random pixel purity index. IEEE Geoscience and Remote Sensing Letters 7(2), 324–328 (2010)
68. Chang, C.L., Wu, C., Tsai, C.: Random N-Finder (N-FINDR) endmember extraction algorithms for hyperspectral imagery. IEEE Transactions on Image Processing 20(3), 641–656 (2011)
69. Rezaei, Y., Mobasheri, M.R., Zoej, M.J.V., Schaepman, M.E.: Endmember extraction using a combination of orthogonal projection and genetic algorithm. IEEE Geoscience and Remote Sensing Letters (in press)
70. Grana, M., Veganzones, M.A.: Endmember induction by lattice associative memories and multi-objective genetic algorithms. EURASIP Journal on Advances in Signal Processing (in press)
71. Duran, O., Petrou, M.: Robust endmember extraction in the presence of anomalies. IEEE Transactions on Geoscience and Remote Sensing 49(6), 1986–1996 (2011)
72. Veganzones, M.A., Grana, M.: A spectral/spatial cbir system for hyperspectral images. IEEE Journal of Selected Topics in Earth Observations and Remote Sensing (in press)
73. Caiafa, C.F., Salerno, E., Proto, A.N., Fiumi, L.: Blind spectral unmixing by local maximization of non-Gaussianity. Signal Processing 88(1), 50–68 (2008)
74. Liu, L., Wang, B., Zhang, L.: Decomposition of mixed pixels based on bayesian self-organizing map and gaussian mixture model. Pattern Recognition Letters 30(9), 820–826 (2009)
75. Liu, L., Wang, B., Zhang, L.: An approach based on self-organizing map and fuzzy membership for decomposition of mixed pixels in hyperspectral imagery. Pattern Recognition Letters 31(11), 1388–1395 (2010)
76. Zhang, B., Sun, X., Gao, L., Yang, L.: Endmember extraction of hyperspectral remote sensing images based on the discrete particle swarm optimization algorithm. IEEE Transactions on Geoscience and Remote Sensing 49(11), 4173–4176 (2011)
77. Zhang, B., Sun, X., Gao, L., Yang, L.: Endmember extraction of hyperspectral remote sensing images based on the ant colony optimization (ACO) algorithm. IEEE Transactions on Geoscience and Remote Sensing 49(7), 2635–2646 (2011)
78. Zare, A., Gader, P., Gurumoorthy, K.S.: Directly measuring material proportions using hyperspectral compressive sensing. IEEE Geoscience and Remote Sensing Letters (in press)

Image Security and Biometrics: A Review

Ion Marqués and Manuel Graña

Grupo de Inteligencia Computacional, UPV/EHU
www.ehu.es/ccwintco

Abstract. Imaging security and biometrics are two heavily connected areas. The quick evolution of biometrics has raised the need of securing biometric data. A majority of this data is visual, which has lead to intensive development of image security techniques for biometric applications. In this paper we give a fast fly over image security approaches and imaging-related biometrics. We present the current state-of-the-art of the interplay between both areas. The emphasis in this paper is the computational methods.

Keywords: Biometrics, Imaging Security, Watermarking, Image Cryptography, Steganography.

1 Introduction

Securing data is an important and evolving area of computer science. Text was the initial asset to be secured. Nowadays visual information is also present in computerized processes. Last years have seen an increasing interest on security methods for image data. The goals can be to ensure:

1. The *authenticity* and/or ownership of the image creator or sender.
2. The *integrity* of the image data, and the ability to know if the image has been altered.
3. *Privacy*, in terms of content and/or ownership of the data.

The developed methods must also usually compel *performance* requirements (speed, memory usage, etc.), *usability* criteria (user-friendliness, no expertise requirements,etc.) and other features that could be necessary. Some security techniques like watermarking and steganography have been present in imaging science for a long time [1]. Adaptation of classic cryptography to image data has also been done [2,3,4]. These areas have recently seen a boost in research interest due to the nature of the image data: Many biometric systems use imaging methods, and the need for secure biometrics storing and sharing schemes is increasing. Figure 1 illustrates the interplay between these two areas.

This paper introduces the cited image security research areas in section 2, citing the most recent developments. Section 3 gives an overview of image-based biometrics and presents how image security techniques are being applied to it. Final conclusions are presented in section 4.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 436–447, 2012.

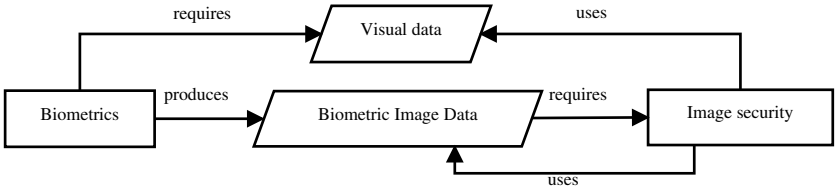


Fig. 1. A simple flowchart that illustrates the relation between biometrics and image security. Note that each arrow involves the use of computational methods.

2 Image Security

Securing the storage and transmission of images is one of the cornerstones of data security. Communication protocols like Secure Sockets Layer (SSL) use Message Authentication Codes to guarantee the correct identities of the sender and receiver of data fragments over the Internet. Similarly, multimedia content such as audio, images or video can be object of authentication, integrity and data hiding procedures. The two main approaches to authentication on imaging science are watermarking and cryptography. The main difference between both methods is that Watermarking aims to introduce the signature of the owner without altering the visual perception of the data. Conversely, encrypted images are not readable without a decryption step. Most watermark methods and cryptosystems also seek data integrity. Image steganography could be seen as a special case of watermarking, where the goal is to hide information into the image.

2.1 Watermarking

The goal of watermarking is to embed data into an image by introducing changes which must comply three requirements: 1) To be imperceptible to the human eye 2) To be recoverable by computer software and 3) To be generated and embedded so that attackers can not have access to it. These requirements and the nature of image data arise some properties that watermarking algorithms must suit [5,6]:

- Fidelity: The higher the fidelity the more difficult is to notice the watermark. This is not a computational feature, but a visual perception subjective measure.
- Capacity: This property corresponds to quantity of information that a watermark can hold.
- Robustness: The watermarking process should be resilient to passive distortion sources. This distortions can be caused by image processing, transmission distortion, and storage distortion. The robustness also corresponds to the ability of the watermarking to resist attacks like watermark removal, covert communication detection, collusion attacks or forgery attacks.

These properties can collide, so the watermarking process must have a proper tradeoff between them. Tremeau et al. gave a good example of these phenomenons [5]. In order to be robust, the watermark should be placed in the most significant parts of the data. In fact, many watermark removal attacks compromise the perceptually less significant components via compression. However, in order to retain a high fidelity, a watermark has to be placed in the less perceptually significant parts of the data. Therefore, robustness and fidelity are in conflict. Its important to know well the application domain of the watermarking process in order to find the right balance between these properties. These applications include [6]:

- Ownership assertion: The owner of an image can generate and embed a unique watermark. The user could make a watermark based on a private key. He or she can not only ensure his or her identity but also claim the ownership of the image, as he or she is the only one that knows the key.
- Data integrity: Any change made to the image will also affect the watermark.
- Fingerprinting: Transactional watermarks allow to link the image data to the receiver of the data. For instance, in a closed or secret media creation process, custom watermarks can help to identify the source of a possible leak. It can also be implemented a copy control system. Instead of preventing illegal copies watermarking can track the illegal activity. Media player and recorders can also be programmed to refuse copying protected material.

Many recent researches focus their goal to a specific domain of image data. Image forgery prevention is one of the areas. Although blind methods are broadly studied [7], watermarks are invaluable tools for image forensics. Another important topic is copyright protection. Many algorithms are designed and tested for a specific video codec or image format [8]. Some applications, like medical imaging or arts storage require that the data cannot be modified -i.e. losses or lossy-to-lossles procedures. Other aspect affecting the development of watermarks is what to encode, for example a 2-Dimensional bar code [9] or a logo [10]. There is also a growing interest in fusing watermark-protected biometric data [11]. Besides well known two dimensional image and video watermarking, research on watermark insertion in three dimensional visual data is been also developed [12]. There has been wide interest in the use of computational intelligence methods for watermarking and are extensively revised in [13]. Some methods use a signal processing approaches like wavelet transforms [14] and Independent Component Analysis [15]. In this line of work, other researches propose fuzzy clustering approaches [16], genetic algorithms [17] or hybrid approaches [18,19]. As an extension of these methods, some researchers seek the capability of retrieving the watermark from the image, in order to test separately its authenticity [20,21]. It is also interesting the ability of not only detecting unwanted modifications but also recovering the original image [22].

2.2 Image Cryptography

The goal of encrypting images is to hide it's content from unauthorized viewers and authenticate its owner. Classic cryptography was centered on text data [23].

Nowadays, more research is being done focused on image data. The idea is to use the visual information as the different components that form a cryptosystem. Furthermore, it is desirable that the procedure does not require additional optical hardware [24]. For example, the amplitude distribution of the Hartley transform can be the public key and the phase distribution the private key [24]. Other similar approaches use Mellin transform [25], Fractional Fourier transform [26] or blind source separation algorithms [27].

Other aspect of cryptography applied to images is Visual Cryptography. The idea is to divide visual information into meaningless trunks and divide them between users. The image can only be reconstructed if all the parts are overlaid in a certain way, hopefully without loss of information [2,3,4]. These methods don't require keys because the human visual system decrypts the data. Visual Cryptography is closely related to Stenography, which is discussed in subsect. 2.3.

2.3 Information Hiding on Images

The science that involves hiding and communicating secret data in a multimedia carrier like images or video is called steganography. Its goal is to hide the very existence of the secret data. This is a key feature in applications like medical image sharing [28] which handle private data. Cheddad et al. [29] published recently an exhaustive survey on image steganography. We will focus on the computational intelligence tools and the latest publications on the matter.

Most algorithms work on spatial [30,31] or frequency [32] domain. They make use of computational tools like predictors [33], particle swarm optimization [34] or fuzzy detectors [35]. Recently, adaptative algorithms are being developed, where more information about the image is used [29,36,37]. The combination of statistical and frequency information with image object or texture knowledge can lead to better results [29]. Some of these approaches even try to enhance the quality of the image at the same time that they embed the data [36]. These techniques are obviously dependent on the image format and aren't usually designed for palette-based images [37].

3 dimensional models can also be subject to steganography. Previous hiding efforts for 3D models were usually modified watermarking techniques. Only since 2009 researchers have started to design 3D steganography algorithms. Chao et al. [38] proposed a multi-layered method. It had high capacity but was not secure against certain malicious attacks such as smoothing, additional noise, nonuniform scaling, simplification, and vertices resampling. In 2010 Amat et al. developed a lossless algorithm where the positions of vertices were not altered [39]. Their method is based on minimum spanning trees. Other recent researches rely on 2D imaging techniques [40].

3 Biometrics and Image Security

The importance of image security is most notable in Biometrics. Biometrics consist on a series of methods for unequivocally recognizing a subject (typically a

human but can also be other animal species). Biometric algorithms and procedures should conform a system which ensures the identity of the target using biological traits: Fingerprint, face image, DNA sequence, voice, walking gaits, etc. Many of these techniques are closely related to imaging science -see table 1. Some methods aim to identify one subject, while others require the verification of the person [41]. Most of biometric systems require strong security. Therefore, they usually make use of watermarking, cryptography and steganography. Biometric systems should have some properties by definition, and also some other issues that must be considered [42,43]:

- Universality: Applicable to every human.
- Distinctiveness: Any two subject's biometric features must be sufficiently distinguishable.
- Permanence: The biometric features should be persistent over time. Obtaining or verifying them should not induce changes in the user's biometric features.
- Collectability: The features can be measured quantitatively.
- Performance: Accuracy, speed, low resource usage and invariability to environmental factors are desirable.
- Acceptance: It is important to measure the social acceptance of a certain biometric characteristic.
- Security: Biometric systems should ensure authenticity, integrity, privacy and resistance to attacks and forgery.

Table 1. Summary of biometric methods and their relationship with imaging techniques. Note: EHF stands for Extremely High Frequency (30-300 GHz wavelength)

Technique	Image-based method? (image type)	Involvement of imaging techniques	
		Acquisition	Verification/identification
Face recognition	Yes (visual)	Yes	Yes
Ear recognition	Yes (visual)	Yes	Yes
Thermography	Yes (infrared)	Yes	Yes
Palmprint/fingerprint	Yes (scan)	Yes	Yes
Iris	Yes (visual)	Yes	Yes
Retinal scan	Yes (infrared)	Yes	Yes
Geometry (e.g. hand)	Yes (scan)	Yes	Yes
Gait	Yes (video)	Yes	Yes
EHF imaging (e.g thorax)	Yes (EHF)	Yes	Yes
Dental	Sometimes	Sometimes	Sometimes
Signature, keystroke	No	Sometimes	Sometimes
Voice	No	No	Sometimes
Odor	No	No	No
DNA	No	No	No

3.1 Imaging and Biometrics

Face recognition [44] is one of the most relevant applications of image analysis. It has been widely proposed and used as a biometric feature. In fact, to build an automated system which equals human ability to recognize faces is one of the core challenges of biometrics. Face recognition may consist in the authentication of a user, which is a binary decision problem. Most commonly, it consists in the search for the identity of a subject in a large face database, which is a (large) multi-class problem. This initial problem can be extended to gaze, expression or mood recognition [45].

Recent researches on this topic have used classic approaches like finding optimal discriminant projections which seek to preserve locality [46],supervised discriminant methods [47] or lattice computing algorithms [48]. Other approach is not to select the optimal features but to have a sufficient number of them via sparsity preserving methods [49,50]. Frequency-based algorithms like wavelet transforms are also being used [51]. Some researches seek to use anthropomorphic geometric features , although this approach has seen fewer interest lately. This approach is an example of what is known as Soft-biometrics. This branch of biometrics uses features "easy" to extract (like skin, eyes, or non-facial features like ethnicity). It can be useful to enhance "hard" face recognition biometric methods [52]. On a side note, the usefulness of infrared or near-infrared imaging for face recognition is still a open issue [53] as the infrared signature of a face can change a lot through time. However, infrared information can help face recognition systems to overcome pose, illumination and expression variations [54].3D information is also being used to build systems invariant to those problems [55].

Unlike face recognition, iris scanners usually require an action by the subject. In other words, the user must come close to the iris scanner and stay still. One of the objectives of iris biometrics is to design less obtrusive acquisition procedures [56]. This is an inconvenience for the user, but prevents problems like occlusion or poor image retrieval. Other advantage is that biometric systems differentiate between left and right eyes and between irises of identical twins [57]. On the other hand, iris biometrics face degradation problems caused by pupil dilation [58] or contact-lenses [59]. Other issue is the segmentation of the iris. An iris recognition system must extract the iris region and discard the pupil, eyelids, sclera, etc. Recent studies have achieved fast and accurate segmentation overcoming reflection problems [60]. The iris texture extraction step is performed using techniques like Discrete Cosine Transform, Fourier Transform, Haar wavelets, Gabor filters, etc. [61,56]. Santos et al. propose in [62] a fusion scheme to take advantage of different extraction techniques. The results show that fusing methods can lead to systems less sensitive to poor quality data. This contribution is relevant in terms of building systems less intrusive.

Other image-based biometric systems include fingerprint or palmprint recognition [63,64,65], hand geometry [66], dental biometrics [67], ear biometrics [68], millimetre-wave scans [69], etc.

Multi-modal biometrics is another current research area. The idea behind the multi-modal or hybrid biometrics is to combine different methods to optimize the aspects listed on the beginning of sec. 3. Some researches develop statistical tools to effectively extract and fuse features from different sources, like face images and palmprint scans [70]. Other recent researches fuse the features at the score or classification level [71,72,73,74]. Computational tools like particle swarm optimization [75] are also being used to enhance the fusion step.

3.2 Biometric Image Security

Biometric data must be appropriately secured, but biometrics also offers a wide array of security applications (e.g. e-passport [41]). However, there are widespread security concerns regarding the stored biometric data. The use of biometric features like face images or fingerprints to enhance classic cryptographic or watermarking systems is a promising approach. This research topic open some concerns: What happens if the biometrics of a subject are stolen? What is the proper balance between performance and robustness? What biometric approach should we use in terms of proper universality, distinctiveness, social acceptance, etc.?

One of the approaches is to secure biometric images via encryption techniques. These methods sometimes perform lossy procedures over the images [76,77]. Generally this systems must decrypt the data in order to proceed to the authentication process. The challenge of bio-cryptography is to implement *cancelable* biometrics [78], which can be described as the application of non-invertible and repeatable modifications to the original biometric templates.

Steganography [34] and watermarking [79,80] are also being employed on biometric data security. This technique allows embedding large amounts of biometric information within an image. Steganography can be employed to embed biometric images into publicly transmitted images [34]. Multimodal biometric image watermarking is also a promising research area [81,11].

4 Conclusion

Computer vision and imaging sciences are closely related to biometrics. The interplay between both research areas is continually evolving. Old biometric systems which relied on human visual verification are being displaced by the superior analyzing capabilities of computers. Image data has become an asset to protect, and we also use imaging techniques to secure data. Thus, new computational advances in steganography, watermarking or pattern recognition boost the development of secure and effective biometric systems. Similarly, ever-growing requirement of trustable biometrics by governments and industry require constant research in such areas.

References

1. Petitcolas, F., Anderson, R., Kuhn, M.: Information hiding - a survey. Proceedings of the IEEE 87(7), 1062–1078 (1999)
2. Yang, C.N., Chen, T.S.: Colored visual cryptography scheme based on additive color mixing. Pattern Recognition 41(10), 3114–3129 (2008)
3. Yang, C.N., Ciou, C.B.: Image secret sharing method with two-decoding-options: Lossless recovery and previewing capability. Image and Vision Computing 28(12), 1600–1610 (2010)
4. Jin, J., Wu, Z.-H.: A secret image sharing based on neighborhood configurations of 2-d cellular automata. Optics & Laser Technology 10.1016/j.optlastec.2011.08.023(0) (2011)
5. Tremeau, A., Muselet, D.: Recent trends in color image watermarking. Journal of Imaging and Science Technology 53(1), 010201 (2009)
6. Cox, I., Miller, M., Bloom, J.: Watermarking applications and their properties. In: Proceedings of International Conference on Information Technology: Coding and Computing, pp. 6–10 (2000)
7. Mahdian, B., Saic, S.: A bibliography on blind methods for identifying image forgery. Signal Processing-Image Communication 25(6), 389–399 (2010)
8. Xu, D., Wang, R., Wang, J.: A novel watermarking scheme for h.264/avc video authentication. Signal Processing-Image Communication 26(6), 267–279 (2011)
9. Kim, J., Kim, N., Lee, D., Park, S., Lee, S.: Watermarking two dimensional data object identifier for authenticated distribution of digital multimedia contents. Signal Processing-Image Communication 25(8), 559–576 (2010)
10. Tsai, H.M., Chang, L.W.: Secure reversible visible image watermarking with authentication. Signal Processing-Image Communication 25(1), 10–17 (2010)
11. Vatsa, M., Singh, R., Noore, A.: Feature based rdwt watermarking for multimodal biometric system. Image and Vision Computing 27(3), 293–304 (2009)
12. Wang, K., Lavoue, G., Denis, F., Baskurt, A.: A comprehensive survey on three-dimensional mesh watermarking. IEEE Transactions on Multimedia 10(8), 1513–1527 (2008)
13. Darwish, A., Abraham, A.: The use of computational intelligence in digital watermarking: Review, challenges, and new trends. Neural Network World 21(4), 277–297 (2011)
14. Cancellaro, M., Battisti, F., Carli, M., Boato, G., Natale, F.D., Neri, A.: A commutative digital image watermarking and encryption method in the tree structured haar transform domain. Signal Processing: Image Communication 26(1), 1–12 (2011)
15. Mostefa, I.B., Braci, S., Delpha, C., Boyer, R., Khamadja, M.: Quantized based image watermarking in an independent domain. Signal Processing-Image Communication 26(3), 194–204 (2011)
16. Chen, W.C., Wang, M.S.: A fuzzy c-means clustering-based fragile watermarking scheme for image authentication. Expert Systems With Applications 36(2), 1300–1307 (2009)
17. Huang, H.C., Chu, C.M., Pan, J.S.: The optimized copyright protection system with genetic watermarking. Soft Computing 13(4), 333–343 (2009); 2nd IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, CA (2006)
18. Deng, C., Gao, X., Li, X., Tao, D.: A local tchebichef moments-based robust image watermarking. Signal Processing 89(8), 1531–1539 (2009)

19. Chang, C.C., Chen, K.N., Lee, C.F., Liu, L.J.: A secure fragile watermarking scheme based on chaos-and-hamming code. Journal of Systems and Software 84(9), 1462–1470 (2011)
20. Chang, C.C., Lin, P.Y.: Adaptive watermark mechanism for rightful ownership protection. Journal of Systems and Software 81(7), 1118–1129 (2008)
21. Cintra, R.J., Dimitrov, V.S., de Oliveira, H.M., Campello de Souza, R.M.: Fragile watermarking using finite field trigonometrical transforms. Signal Processing-Image Communication 24(7), 587–597 (2009)
22. Zhang, X., Wang, S.: Fragile watermarking scheme using a hierarchical mechanism. Signal Processing 89(4), 675–679 (2009)
23. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory 22, 644–654 (1976)
24. Hwang, H.-E.: An optical image cryptosystem based on hartley transform in the fresnel transform domain. Optics Communications 284(13), 3243–3247 (2011)
25. Zhou, N., Wang, Y., Wu, J.: Image encryption algorithm based on the multi-order discrete fractional mellin transform. Optics Communications 284(24), 5588–5597 (2011)
26. Zhong, Z., Chang, J., Shan, M., Hao, B.: Fractional fourier-domain random encoding and pixel scrambling technique for double image encryption. Optics Communications 285(1), 18–23 (2012)
27. Lin, Q.H., Yin, F.L., Mei, T.M., Liang, H.: A blind source separation-based method for multiple images encryption. Image and Vision Computing 26(6), 788–798 (2008)
28. Ulutas, M., Ulutas, G., Nabiyev, V.V.: Medical image security and epr hiding using shamir's secret sharing scheme. Journal of Systems and Software 84(3), 341–353 (2011)
29. Cheddad, A., Condell, J., Curran, K., Kevitt, P.M.: Digital image steganography: Survey and analysis of current methods. Signal Processing 90(3), 727–752 (2010)
30. Kim, K.S., Lee, M.J., Lee, H.Y., Lee, H.K.: Reversible data hiding exploiting spatial correlation between sub-sampled images. Pattern Recognition 42(11), 3083–3096 (2009)
31. Tai, W.L., Yeh, C.M., Chang, C.C.: Reversible data hiding based on histogram modification of pixel differences. IEEE Transactions on Circuits and Systems for Video Technology 19(6), 904–908 (2009)
32. Braci, S., Delpha, C., Boyer, R.: How quantization based schemes can be used in image steganographic context. Signal Processing-Image Communication 26(8-9), 567–576 (2011)
33. Tseng, H.W., Hsieh, C.P.: Prediction-based reversible data hiding. Information Sciences 179(14), 2460–2469 (2009)
34. Qi, M., Lu, Y., Du, N., Zhang, Y., Wang, C., Kong, J.: A novel image hiding approach based on correlation analysis for secure multimodal biometrics. Journal of Network and Computer Applications 33(3), 247–257 (2010)
35. Chang, C.C., Lee, J.S., Le, T.H.N.: Hybrid wet paper coding mechanism for steganography employing n-indicator and fuzzy edge detector. Digital Signal Processing 20(4), 1286–1307 (2010)
36. Wu, C.C., Kao, S.J., Hwang, M.S.: A high quality image sharing with steganography and adaptive authentication scheme. Journal of Systems and Software 84(12), 2196–2207 (2011)
37. Zhao, H., Wang, H., Khan, M.K.: Steganalysis for palette-based images using generalized difference image and color correlogram. Signal Processing 91(11), 2595–2605 (2011)

38. Chao, M.W., Lin, C.H., Yu, C.W., Lee, T.Y.: A high capacity 3d steganography algorithm. IEEE Transactions on Visualization and Computer Graphics 15(2), 274–284 (2009)
39. Amat, P., Puech, W., Druon, S., Pedeboy, J.P.: Lossless 3d steganography based on mst and connectivity modification. Signal Processing-Image Communication 25(6), 400–412 (2010)
40. Elsheh, E., Hamza, A.B.: Secret sharing approaches for 3d object encryption. Expert Systems with Applications 38(11), 13906–13911 (2011)
41. Schouten, B., Jacobs, B.: Biometrics and their use in e-passports. Image and Vision Computing 27(3), 305–312 (2009)
42. Jain, A., Ross, A., Prabhakar, S.: An introduction to biometric recognition. IEEE Transactions on Circuits and Systems for Video Technology 14(1), 4–20 (2004)
43. Biometric technology today, vol. 2011 (2011)
44. Chellappa, R., Sinha, P., Phillips, P.J.: Face recognition by computers and humans. IEEE Computer 43(2), 46–55 (2010)
45. Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on local binary patterns: A comprehensive study. Image and Vision Computing 27(6), 803–816 (2009)
46. Gui, J., Jia, W., Zhu, L., Wang, S.L., Huang, D.S.: Locality preserving discriminant projections for face and palmprint recognition. Neurocomputing 73(13-15), 2696–2707 (2010)
47. Wan, M., Lai, Z., Shao, J., Jin, Z.: Two-dimensional local graph embedding discriminant analysis (2dlgeda) with its application to face and palm biometrics. Neurocomputing 73(1-3), 197–203 (2009)
48. Marques, I., Graña, M.: Face recognition with lattice independent component analysis and extreme learning machines. Soft Computing (in press)
49. Qiao, L., Chen, S., Tan, X.: Sparsity preserving projections with applications to face recognition. Pattern Recognition 43(1), 331–341 (2010)
50. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2), 210–227 (2009)
51. Zhang, T., Fang, B., Yuan, Y., Tang, Y.Y., Shang, Z., Li, D., Lang, F.: Multiscale facial structure representation for face recognition under varying illumination. Pattern Recognition 42(2), 251–258 (2009)
52. Marcialis, G.L., Roli, F., Muntoni, D.: Group-specific face verification using soft biometrics. Journal of Visual Languages and Computing 20(2), 101–109 (2009)
53. Hollingsworth, K., Bowyer, K.W., Flynn, P.J.: Useful features for human verification in near-infrared periocular images. Image and Vision Computing 10.1016/j.imavis.2011.09.002(0) (2011)
54. Pan, Z., Healey, G., Prasad, M., Tromberg, B.: Face recognition in hyperspectral images. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(12), 1552–1560 (2003)
55. Efraty, B., Bilgazyev, E., Shah, S., Kakadiaris, I.A.: Profile-based 3d-aided face recognition. Pattern Recognition 45(1), 43–53 (2012)
56. Bowyer, K.W., Hollingsworth, K., Flynn, P.J.: Image understanding for iris biometrics: A survey. Computer Vision and Image Understanding 110(2), 281–307 (2008)
57. Hollingsworth, K., Bowyer, K.W., Lagree, S., Fenker, S.P., Flynn, P.J.: Genetically identical irises have texture similarity that is not detected by iris biometrics. Computer Vision and Image Understanding 115(11), 1493–1502 (2011)

58. Hollingsworth, K., Bowyer, K.W., Flynn, P.J.: Pupil dilation degrades iris biometric performance. Computer Vision and Image Understanding 113(1), 150–157 (2009)
59. Baker, S.E., Hentz, A., Bowyer, K.W., Flynn, P.J.: Degradation of iris recognition performance due to non-cosmetic prescription contact lenses. Computer Vision and Image Understanding 114(9), 1030–1044 (2010)
60. He, Z., Tan, T., Sun, Z., Qiu, X.: Toward accurate and fast iris segmentation for iris biometrics. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(9), 1670–1684 (2009)
61. Kumar, A., Passi, A.: Comparison and combination of iris matchers for reliable personal authentication. Pattern Recognition 43(3), 1016–1026 (2010)
62. Santos, G., Hoyle, E.: A fusion approach to unconstrained iris recognition. Pattern Recognition Letters 10.1016/j.patrec.2011.08.017(0) (2011)
63. Amayeh, G., Bebis, G., Erol, A., Nicolescu, M.: Hand-based verification and identification using palm-finger segmentation and fusion. Computer Vision and Image Understanding 113(4), 477–501 (2009)
64. Chen, J., Moon, Y.S., Wong, M.F., Su, G.: Palmprint authentication using a symbolic representation of images. Image and Vision Computing 28(3), 343–351 (2010)
65. Kong, A., Zhang, D., Kamel, M.: A survey of palmprint recognition. Pattern Recognition 42(7), 1408–1418 (2009)
66. Nicolae, D.: A survey of biometric technology based on hand shape. Pattern Recognition 42(11), 2797–2806 (2009)
67. Lin, P.L., Lai, Y.H., Huang, P.W.: Dental biometrics: Human identification based on teeth and dental works in bitewing radiographs. Pattern Recognition 45(3), 934–946 (2012)
68. Arbab-Zavar, B., Nixon, M.S.: On guided model-based analysis for ear biometrics. Computer Vision and Image Understanding 115(4), 487–502 (2011)
69. Alefs, B., den Hollander, R., Nennie, F., van der Houwen, E., Bruijn, M., van der Mark, W., Noordam, J.: Thorax biometrics from millimetre-wave images. Pattern Recognition Letters 31(15), 2357–2363 (2010)
70. Xu, Y., Zhang, D., Yang, J.Y.: A feature extraction method for use with bimodal biometrics. Pattern Recognition 43(3), 1106–1115 (2010)
71. Xu, Y., Zhu, Q., Zhang, D.: Combine crossing matching scores with conventional matching scores for bimodal biometrics and face and palmprint recognition experiments. Neurocomputing 74(18), 3946–3952 (2011)
72. Marcialis, G.L., Roli, F., Didaci, L.: Personal identity verification by serial fusion of fingerprint and face matchers. Pattern Recognition 42(11), 2807–2817 (2009)
73. Alsaade, F., Ariyaeeinia, A., Malegaonkar, A., Pillay, S.: Qualitative fusion of normalised scores in multimodal biometrics. Pattern Recognition Letters 30(5), 564–569 (2009)
74. Hanmandlu, M., Grover, J., Gureja, A., Gupta, H.: Score level fusion of multimodal biometrics using triangular norms. Pattern Recognition Letters 32(14), 1843–1850 (2011)
75. Raghavendra, R., Dorizzi, B., Rao, A., Kumar, G.H.: Designing efficient fusion schemes for multimodal biometric systems using face and palmprint. Pattern Recognition 44(5), 1076–1088 (2011)
76. Bhatnagar, G., Wu, J., Raman, B.: Fractional dual tree complex wavelet transform and its application to biometric security during communication and transmission. Future Generation Computer Systems 28(1), 254–267 (2012)

77. Acharya, B., Sharma, M.D., Tiwari, S., Minz, V.K.: Privacy protection of biometric traits using modified hill cipher with involutory key and robust cryptosystem. Procedia Computer Science 2(0), 242–247 (2010)
78. Bolle, R.M., Connell, J.H., Ratha, N.K.: Biometric perils and patches. Pattern Recognition 35(12), 2727–2738 (2002)
79. Lee, H., Lim, J., Yu, S., Kim, S., Lee, S.: Biometric image authentication using watermarking. In: International Joint Conference on SICE-ICASE 2006, pp. 3950 –3953 (2006)
80. Allah, M.M.A.: Embedded biometric data for a secure authentication watermarking. In: Proceedings of the Fourth conference on IASTED International Conference: Signal Processing, Pattern Recognition, and Applications, Anaheim, CA, USA, pp. 191–196. ACTA Press (2007)
81. Kim, W.G., Lee, H.: Multimodal biometric image watermarking using two-stage integrity verification. Signal Processing 89(12), 2385–2399 (2009)

Cocaine Dependent Classification Using Brain Magnetic Resonance Imaging

M. Termenon[1], Manuel Graña[1], A. Barrós-Loscertales[2], J.C. Bustamante[2], and C. Ávila[2]

[1] Grupo de Inteligencia Computacional (GIC), UPV/EHU
[2] Dpto. Psicología Básica, Clínica y Psicobiología, Universitat Jaume I, Castellón de la Plana, Spain
www.ehu.es/ccwintco

Abstract. The purpose of this study is to elucidate if it is possible to discriminate between cocaine dependent patients and healthy controls applying computer aided diagnosis tools to brain magnetic resonance imaging. Feature extraction was done computing Pearson's correlation using subjects class as indicative variable. Linear support vector machines classifiers were trained and tested on the most significative voxels using leave one out cross-validation process. Results show that classifier achieve on average almost perfect accuracy, sensitivity and specificity in a group of 30 cocaine-dependent and 35 controls, supporting the usefulness of this process to discriminate between these subjects.

Keywords: MRI, Cocaine, SVM.

1 Introduction

Computer Aided Diagnosis (CAD) tools are playing an increasingly role in analyzing medical images such as magnetic resonance imaging (MRI), computed tomography (CT) or Single-photon emission computed tomography (SPECT). These tools apply machine learning (ML) and pattern recognition (PR) techniques to obtain significative statistical differences in images that discriminate between diseases or different stages of a disease. Focusing on brain imaging, there are several studies that have used these ML and PR techniques to classify and localize several neuropsychiatric diseases: Alzheimer disease [1,2], schizophrenia [3], bipolar disorder [4] and so on.

In this work, we are going to focus in the neuropsychiatric complications derived from the cocaine abuse. Cocaine is one of the most consumed illegal drugs and its chronic abuse may cause consequences such as ischemic, hemorrhagic strokes, depression and neuropsychological abnormalities [5]. Selected regions in the brain of cocaine consumers show functional, neurochemical and structural abnormalities that can be used to identify the differences between the brains of cocaine consumers and non-consumers and then, to select an adecuate pharmacotherapy to treat this disorder [6].

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 448–454, 2012.

Studies found structural differences in striatum, frontal gyrus, parahippocampus, posterior cingulate, amygdala, insula and cerebellum [7], ventromedial orbitofrontal, anterior cingulate, anteroventral insular and superior temporal cortices [6]. Functional MRI (fMRI) tests assert that chronic cocaine consumption may affect the attentional system in the right parietal lobe, making patients more prone to attention deficits [8] and detect reductions in primary visual cortex and primary motor cortex after cocaine administration [9]. In white matter (WM), reduced fractional anisotrophy (FA) in the genu and rostral body of the anterior corpus callosum in cocaine-dependent subjects compared to controls [10] and also lower FA and higher mean diffusivity in frontal and parietal WM regions [11]. Lim et al [12] suggested that duration of cocaine use was associated with decreased grey and white matter volumes. A multimodal study [5] showed that cocaine dependent patients have generalized cerebral hypoperfusion.

Classification techniques were applied to detect prenatal cocaine exposure in adolescents [13] using nonlinear support vector machines (SVMs) on features extracted from structural and functional MRI images. Also, majority vote classification technique [14] was applied using fMRI for cocaine addicted and control subjects. For the time being, classification techniques related to cocaine addiction have only been applied combining functional and structural images, which implies higher dimensionality, noise level and subject variability than working only structural images. In this paper, we present a simple, well grounded methodology, that extracts significant information from MRI-T1 structural images and use that information to discriminate between cocaine consumers and non-consumers.

In the next section 2, methods and materials used for this experiment will be described. Section 3 shows the experimental results and finally, in section 4, conclusion obtained from results are stablished.

2 Methods and Materials

Thirty cocaine-dependent male patients (mean = 34.41 ± 6.62) and thirty-five matched controls (mean = 33.38 ± 7.87) participated in this study. The cocaine patients were recruited from the Addiction Treatment Service of San Agustín in Castellón, Spain. The inclusion criteria for cocaine dependence was based on the DSM-IV criteria. Control subjects were required to have no diagnosis of substance abuse or dependence. The exclusion criteria for all the participants included neurological illness, prior head trauma, positive HIV status, diabetes, Hepatitis C, or other medical illness and psychiatric disorders. Cocaine consumption was assessed with an urine toxicology test, which ensured a minimum period of abstinence of two to four days prior to MRI data acquisition. Groups were matched on the basis of age and level of education. All the participants were right-handed according to the Edinburgh Handedness Inventory (Oldfield, 1971; Bryden, 1977). They all signed an informed consent prior to participating in this study.

Images were acquired on a 1.5T Siemens Avanto (Erlangen, Germany) with a standard quadrature head coil. A high resolution 3D T1-weighted gradient

echo pulse sequence was acquired (TE=4.9 ms; TR=11 ms; FOV=24 cm; matrix=256×224×176; voxel size=1×1×1).

Appropriate data preprocessing, ensuring anatomical correspondence of voxels intersubjects, is of paramount importance to obtain the optimal performance of feature extraction and classification processes. Images were preprocessed with FSL software (http://www.fmrib.ox.ac.uk/fsl/). T1-weighted sMRI volumes were skull stripped, reoriented and affine registered to the Montreal Neurological Institute (MNI152) standard template. Then, three resolution nonlinear diffeomorphic registration (fast symmetric normalization) of affine registered data [15] to MNI152 template was computed using ANTS software (http://www.picsl.upenn.edu/ANTS). This spatial normalization was done to ensure the correspondence between voxel sites and anatomical features across all subjects.

Feature extraction process was performed computing a voxel-wise Pearson's correlation with the indicator variable specifying the subject class label. Computing the empirical distribution of the correlation coefficients, we select voxels with high absolute correlation belonging to a certain percentile of this distribution. Gray values of these selected voxels were then classified using a linear kernel support vector machines classifier. Feature extraction pipeline is shown in Figure 1.

2.1 Support Vector Machines

The Support Vector Machines (SVMs) [16,17] approach is a PR technique based on the statistical learning theory. Its training principle consists of finding the optimal linear hyperplane that minimize the expected classification error. The classification approach works to solve the following optimization problem:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i, \tag{1}$$

subject to

$$y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq (1-\xi_i),\ \xi_i \geq 0,\ i = 1, 2, \ldots, n. \tag{2}$$

The minimization problem is solved via its dual optimization problem:

$$\min_{\alpha} \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha}, \tag{3}$$

subject to

$$\mathbf{y}^T\boldsymbol{\alpha} = 0,\ 0 \leq \alpha_i \leq C,\ i = 1, \ldots, l. \tag{4}$$

Where $\mathbf{e}$ is a vector of all ones, $C > 0$ is the upper bound on the error and Q is an $l \times l$ positive semidefinite matrix. The Q elements are based on a kernel function, $K(\mathbf{x}_i, \mathbf{x}_j)$, that describes the behavior of the support vectors. The chosen kernel function results in different kinds of SVM with different performance levels, and the choice of the appropriate kernel for a specific application is a difficult task. In this study we only needed to use a linear kernel, defined as:

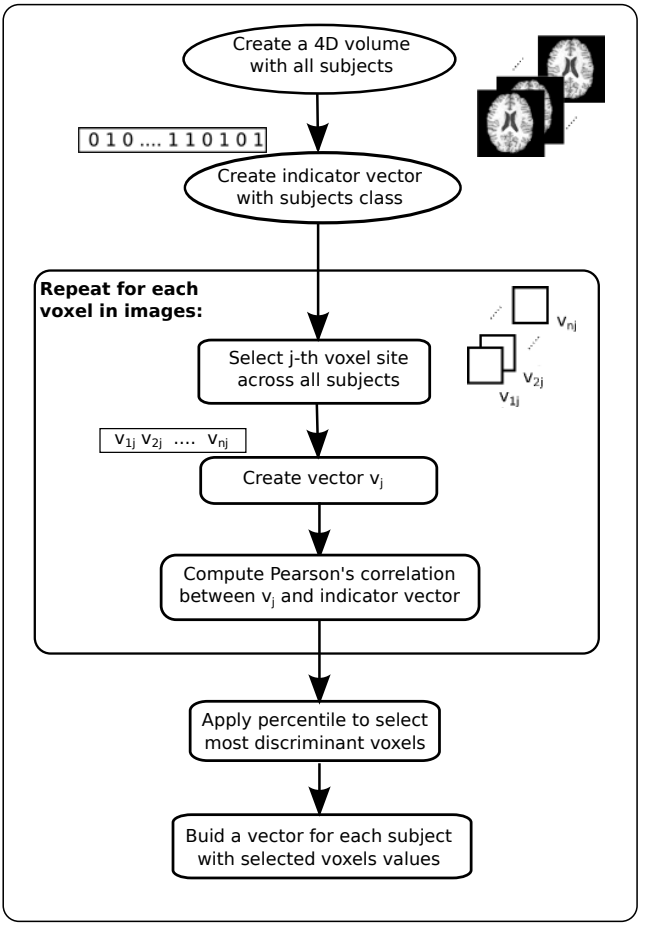


Fig. 1. Features extraction pipeline

$$K(\mathbf{x}_i, \mathbf{x}_j) = 1 + \mathbf{x}_i^T \mathbf{x}_j. \tag{5}$$

This kernel shows good performance for linearly separable data.

3 Results

The average values of the performance measures obtained from leave one out cross-validation steps for some specific values of correlation distribution percentiles are shown in Table 1. We select as classifier SVM with linear kernel. Results are so good that we do not need to consider using a non linear kernel. We test the algorithm with 15 different percentiles corresponding to different number of features, as shown in Table 2.

Table 1. SVM classification results

(%)	**99.50 - 99.90 (steps 0.05)**	**99.92**	**99.95**	**99.97**	**99.99**	**99.995**	**99.999**
Specificity	100.00	100.00	100.00	100.00	100.00	96.67	80.00
Sensitivity	100.00	100.00	100.00	100.00	97.14	97.14	88.57
Accuracy	100.00	100.00	100.00	100.00	98.46	96.92	84.61

Table 2. Number of features depending of the choosen percentile

Percentile (%)	**# Features**
99.50	10,624
99.55	9,561
99.60	8,499
99.65	7,437
99.70	6,374
99.75	5,312
99.80	4,250
99.85	3,187
99.90	2,125
99.92	1,699
99.95	1,062
99.97	637
99.99	212
99.995	106
99.999	21

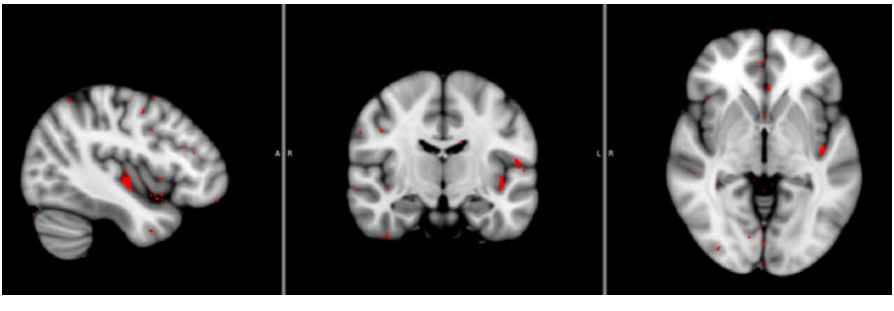


Fig. 2. Most discriminant voxels

Most significant voxels, shown in Figure 2, were found in cerebral cortex, planum polare, insula, parahippocampus and cerebellum. To obtain features localization, we used the AtlasQuery tool of FSL, analizing three different atlases: MNI structural atlas and Harvard-Oxford cortical and subcortical atlases. For the percentile 99.50%, discriminant voxels are shown in red on MNI template.

4 Conclusion

In this paper, we present a procedure for the construction of classifiers able to distinguish with 100% of accuracy cocaine dependent patients from controls using structural brain magnetic resonance imaging. We preprocess the images to ensure anatomical correspondence of voxels intersubjects, extract the most significant features (applying Pearson's correlation) and use machine learning algorithms to classify these features.

Results are 100% accuracy, sensitivity and specificity for almost all the percentiles we tested. Even reducing the number of features to 21, classification results are still good. We found significant information in cerebral cortex, planum polare, insula, parahippocampus and cerebellum. These findings are also found in the literature, supporting our methodology and validating our results.

The recruitment of new cocaine dependent patients and healthy controls data is required to allow for a more extensive proof of the approach.

References

1. Graña, M., Termenon, M., Savio, A., Gonzalez-Pinto, A., Echeveste, J., Pérez, J.M., Besga, A.: Computer aided diagnosis system for alzheimer disease using brain diffusion tensor imaging features selected by pearson's correlation. Neuroscience Letters 502(3), 225–229 (2011) PMID: 21839143
2. Savio, A., García-Sebastián, M., Chyzyk, D., Hernandez, C., Graña, M., Sistiaga, A., de Munain, A.L., Villanúa, J.: Neurocognitive disorder detection based on feature vectors extracted from VBM analysis of structural MRI. Computers in Biology and Medicine (2011)
3. Savio, A., Charpentier, J., Termenon, M., Shinn, A.K., Graña, M.: Neural classifiers for schizophrenia diagnostic support on diffusion imaging data. Neural Network World 20, 935–949 (2010)
4. Frangou, S.: CS02-02 - risk and resilience markers: Use of whole-brain structural MR scans to predict familial risk and disease expression in bipolar disorder. European Psychiatry 26(suppl.1) (2011)
5. Ernst, T., Chang, L., Oropilla, G., Gustavson, A., Speck, O.: Cerebral perfusion abnormalities in abstinent cocaine abusers: a perfusion MRI and SPECT study. Psychiatry Research: Neuroimaging 99(2), 63–74 (2000)
6. Franklin, T.R., Acton, P.D., Maldjian, J.A., Gray, J.D., Croft, J.R., Dackis, C.A., O'Brien, C.P., Childress, A.R.: Decreased gray matter concentration in the insular, orbitofrontal, cingulate, and temporal cortices of cocaine patients. Biological Psychiatry 51(2), 134–142 (2002) PMID: 11822992
7. Barrós-Loscertales, A., Garavan, H., Bustamante, J.C., Ventura-Campos, N., Llopis, J.J., Belloch, V., Parcet, M.A., Ávila, C.: Reduced striatal volume in cocaine-dependent patients. NeuroImage 56(3), 1021–1026 (2011)
8. Bustamante, J.C., Barrós-Loscertales, A., Ventura-Campos, N., Sanjún, A., Llopis, J.J., Parcet, M.A., Ávila, C.: Right parietal hypoactivation in a cocaine-dependent group during a verbal working memory task. Brain Research 1375, 111–119 (2011)
9. Li, S., Biswal, B., Li, Z., Risinger, R., Rainey, C., Cho, J., Salmeron, B.J., Stein, E.A.: Cocaine administration decreases functional connectivity in human primary visual and motor cortex as detected by functional MRI. Magnetic Resonance in Medicine 43(1), 45–51 (2000)

10. Moeller, F.G., Hasan, K.M., Steinberg, J.L., Kramer, L.A., Dougherty, D.M., Santos, R.M., Valdes, I., Swann, A.C., Barratt, E.S., Narayana, P.A.: Reduced anterior corpus callosum white matter integrity is related to increased impulsivity and reduced discriminability in Cocaine-Dependent subjects: Diffusion tensor imaging. Neuropsychopharmacology 30(3), 610–617 (2004)
11. Lane, S.D., Steinberg, J.L., Ma, L., Hasan, K.M., Kramer, L.A., Zuniga, E.A., Narayana, P.A., Moeller, F.G.: Diffusion tensor imaging and decision making in cocaine dependence. PloS One 5(7), e11591 (2010) PMID: 20661285
12. Lim, K.O., Wozniak, J.R., Mueller, B.A., Franc, D.T., Specker, S.M., Rodriguez, C.P., Silverman, A.B., Rotrosen, J.P.: Brain macrostructural and microstructural abnormalities in cocaine dependence. Drug and Alcohol Dependence 92(1-3), 164–172 (2008), PMID: 17904770 PMCID: 2693223
13. Fan, Y., Rao, H., Hurt, H., Giannetta, J., Korczykowski, M., Shera, D., Avants, B.B., Gee, J.C., Wang, J., Shen, D.: Multivariate examination of brain abnormality using both structural and functional MRI. NeuroImage 36(4), 1189–1199 (2007) PMID: 17512218
14. Honorio, J., Samaras, D., Tomasi, D., Goldstein, R.: Simple fully automated group classification on brain fMRI, pp. 1145–1148 (April 2010)
15. Avants, B., Epstein, C., Grossman, M., Gee, J.: Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. Medical Image Analysis 12(1), 26–41 (2008)
16. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience (September 1998)
17. Noble, W.S.: What is a support vector machine? Nat. Biotech. 24(12), 1565–1567 (2006)

A Non-parametric Approach for Accurate Contextual Classification of LIDAR and Imagery Data Fusion

Jorge Garcia-Gutierrez, Daniel Mateos-Garcia, and Jose C. Riquelme-Santos

Department of Computer Languages and Systems, University of Seville,
Reina Mercedes s/n, 41012 Seville, Spain
{jorgarcia,mateosg,riquelme}@us.es

Abstract. Light Detection and Ranging (LIDAR) has become a very important tool to many environmental applications. This work proposes to use LIDAR and image data fusion to develop high-resolution thematic maps. A novel methodology is presented which starts building a matrix of statistics from spectral and spatial information by feature extraction on the available bands (RGB from images, and intensity and height from LIDAR). Then, a contextual classification is applied to generate the final map using a support vector machine (SVM) to classify every cell and the nearest neighbor (NN) rule to sequentially reclassify each cell. The results obtained by this novel method, called SVMNNS (SVM and NN Stacking), are compared with non-contextual and contextual SVMs. It is shown that SVMNNS obtains the best results when applied to real data from the Iberian peninsula.

Keywords: Remote sensing, supervised learning, contextual classifiers.

1 Introduction

Light Detection and Ranging technology (LIDAR) is a remote sensing laser-based technology that, as a main characteristic, can determine the distance from the source to an object or surface providing, not only the x-y position, but also the coordinate z for every impact. The distance to the object is determined by measuring the time between the pulse emission and the detection of the reflected signal, taking into account the position of the emitter. The main applications of LIDAR are related to digital elevation model extraction [2], forest inventories [1] and fuel models [5].

Thematic maps are remote sensing products used to study and manage geographical areas of interest according to a special theme. Although thematic maps have traditionally been generated from aerial and satellite images, the appearance of new sensors has led to increasing interest in data fusion to obtain high resolution products. In this way, LIDAR data-fused thematic maps can be seen in literature to map fire risk [14] or plant communities relations [17].

Classification techniques are traditionally applied to generate thematic maps. If the information about each element's neighbour in addition to the element

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 455–466, 2012.

itself is used, the classification process is said to be contextual [4]. One of the latest examples of a contextual classifier can be seen in [15]. The authors propose a technique called SVMMRF for hyperspectral classification consisting of two steps. In the first step, a probabilistic support vector machine (SVM) is applied to a pixelwise classification of hyperspectral images. In the second step, spatial contextual information is used for refining the classification results with a Markov random field (MRF) regularization.

In this study, we show a novel methodology to develop high resolution maps from LIDAR and imagery data fusion. This methodology is based on a contextual classifier technique for data fusion, we have called SVMNNS (SVM and Nearest Neighbour Stacking). SVMNNS tries to improve the contextual regularization with a non-parametric technique such as the nearest neighbour rule used in previous methodologies [8] but in an iterative manner which is the main difference with our previous approaches [7]. In addition, a fair comparison with another cutting-edge contextual technique (not addressed in our previous work) is shown when a high resolution classification is required.

This article is organized in the following manner. Section 2 provides a description of the methodology. Section 3 shows and analyzes the results achieved. Finally, Section 4 presents the main conclusions of the study and suggests future lines of investigation.

2 Method

This article proposes a novel method, called SVMNNS, which is contextual. The method has the objective of generating thematic maps from data fusion, supported by the use of several families of classifiers (SVM and Nearest Neighbour). Each SVMNNS step is described in detail in the following subsections.

2.1 Data Description

To carry out this study, data were collected from two zones of the Iberian Peninsula (see Figure 1). LIDAR and orthophotography data were obtained from each zone. In both cases, the altitudes of the LIDAR data were normalized through a digital elevation model generated by morphological filters on the LIDAR data [9]. Then, to make the LIDAR intensities usable, it was also necessary to perform normalization [11]. However, due to the scarce density of returns per square meter, the pulses with multiple impacts were not eliminated.

Orthophotographies of both areas were provided by the Regional Ministry of Andalusia in the case of Huelva, and the Laborate Group of the University of Santiago in the case of Trabada and they were taken in 2007 and 2004, respectively. In the case of Huelva, images have a resolution of 0.2 m whilst Trabada images have a resolution of 0.5 m. In both cases, they just contained RGB information.

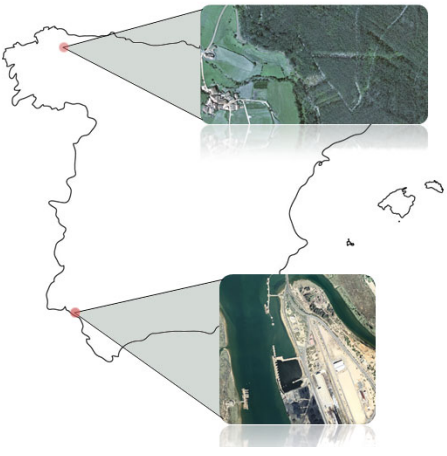


Fig. 1. Study areas of Trabada in the north and Huelva in the south

2.2 Feature Set Extraction

The first step in creating a thematic map is the generation of a matrix that covers the surface to be classified. In our case, the matrix has a vector of features associated to each of its cells which are calculated from images and LIDAR pulses. This fact involves selecting a resolution to set the size of cell in the matrix. The resolution is closely related to the density of the LIDAR pulses. In the case of Huelva, because the data had a low pulse density, the resolution value was set at 3 m^2, whereas in Trabada, where the data had a greater density, the resolution was set at 1 m^2.

In the next step, the features associated with each cell of the matrix are calculated. Seventy-one features were defined based on various other studies [6]. In Table 1, 54 features (9 measures for each of the 6 bands) are shown that are calculated from the available bands: RGB from the images, LIDAR intensity, LIDAR normalized altitudes and the SNDVI (Simulated Normalized Difference Vegetation Index). The SNDVI band simulates the NDVI (Normalized Difference Vegetation Index), substituting the value of the infrared for a value of the LIDAR return intensity. In addition, 17 other features were defined from the distributions of the LIDAR impacts (Table 2).

2.3 Training Set Selection

With a feature matrix calculated, training data are generated. A set of cells of the matrix is classified assigning them a label according to their land cover through a visual inspection of the aerial images or from other data sources. We selected 618 and 332 cells in the case of Huelva and Trabada, respectively. In both cases, the quantity of training instances was approximately 1% of the total. Every instance for the classification is then formed by the vector of stacked textures described in the previous subsection and a label that identifies its class.

Table 1. Band-based texture set

Symbol	Description	Symbol	Description
MAX	Maximum	MIN	Minimum
RANG	Range	STD	Standard Deviation
VAR	Variance	MEAN	Mean
KURT	Kurtosis	CV	Coefficient of Variation
SKEW	Skewness		

Table 2. LIDAR-distribution-based texture set

Symbol	Description
PCTR1	Percentage of first return
NOTFIRST	Number of non-first impacts
PCTR2	Percentage of second return
NEMP	Number of empty neighbours
PCTR3	Percentage of third or later return
TOTALR	Total number of impacts
PCTR21	PCTR2 over PCTR1
PEC	Penetration coefficient
PCTR31	PCTR3 over PCTR1
IID	Difference among first and last impacts in the pixel
PCTR32	PCTR3 over PCTR2
PCTN1	Percentage of single impact
EXTRASLP	Slope among every neighbor
PCTN2	Percentage of double impact
INTRASLP	Slope in the pixel
PCTN3	Percentage of triple or more impact
CRR	Canopy Relief Ratio

Table 3. Parameters C and γ for the radial basis function kernel applied to the SMO algorithm in each area

Area	C	γ
Huelva	8.12	6.17
Trabada	7.93	4.15

As is advised in various studies [18], a pre-process should be applied to data before generating a classification model. Thus, three types of filters are used. First, the missing values are replaced by the corresponding mean value. Second, the data are normalized. Finally, to avoid problems with dimensionality, a feature selection based on correlations is applied. The three filters are executed in Weka

[10] with default parameters. Once the filtered database has been obtained, the next phase is the execution of the SVMNNS algorithm.

2.4 SVMNNS

SVMNNS is characterized by applying two levels of classification by means of a stacking of two well-known classifiers: SVM and k-nearest neighbor (k-NN). Thus, from the training data, an SVM is initially generated by the SMO optimization algorithm [13] implemented in Weka (which was optimized by evolutionary computation [12] with the parameters in Table 3). The SVM performs a first-level classification for every cell of the matrix, assigning a value to its labels.

The next step is the application of a k-NN to reclassify each cell. The value of k is a parameter that the user has to set up at the beginning of the execution. This second level of classification is carried out with a implementation of the k-NN algorithm (IBk in Weka). For each cell, the k-NN is trained with just the values of the features of the adjacent neighbours in the matrix. Thus, SVMNNS introduces the context of each instance in its own classification.

For the zones that were used in this study, it was shown through experimental analysis that the best results were obtained with an 8-adjacency and a value of $k = 7$ (after testing on $k = 3, k = 5, k = 7$), that is, each NN was based on the 7 nearest neighbors from the 8 adjacent elements of every pixel. It should be kept in mind that the k-NN algorithm is very fast for a small number of training instances T. In our case, T depends on the number of adjacent elements where $T <= 8$ is satisfied. Knowing that for all k-NN, $k <= T$ is satisfied, we can conclude that the method is computationally tractable.

The number of refinements, which can be seen as a number of iterations, is also a configurable parameter n. The value of n is also defined by the user at the beginning of the execution of the SVMNNS method. With regard to the areas of this study, the number of iterations was set at 6 to assure the convergence which is reached at the fifth iteration in classical Markovian contextual classifiers [4].

Finally, after going through the final iteration, the algorithm generates a thematic map in which each pixel is associated with the label of the corresponding cell in the same position in the matrix.

3 Experiments

3.1 Results

To confirm the quality of the SVMNNS method, a comparison was established with two other classifiers: SVMs and SVMMRFs. To make a fair comparison of the results, the contextual algorithms (SVMNNS, SVMMRF) used the same initial classification obtained by the non-contextual SVM (optimized by the SMO algorithm) and the same number of subsequent refinements. The comparison is founded on a well-known testing strategies in remote sensing: a hold-out process.

Table 4. Hold-out percentage results for Huelva

		SVM		SVMMRF		SVMNNS	
Class	Num. instan.	Comm. error	Omiss. error	Comm. error	Omiss. error	Comm. error	Omiss. error
Water	2151	3.00	4.10	0.00	3.50	0.00	0.30
Marsh	1266	14.60	27.50	3.00	25.50	3.50	18,40
Roads	1083	23.30	10.70	6.40	4.00	4.50	4,30
Low Veg.	686	24.10	18.10	20.80	9.30	14.40	11.10
Mid. Veg.	464	49.80	48.50	29.50	13.50	30.20	11.20
High Veg.	329	37.10	42.50	19.50	21.40	4.30	12.50
Buildings	1314	20.20	22.90	11.70	4.80	4.10	3.40
Landfills	209	66.50	27.10	81.80	0.00	63.60	0.00
KIA		0.77		0.87		0.91	
Accuracy		81.02		89.66		92.90	

Table 5. Per-class partial accuracies for Huelva

	SVM		SVMMRF		SVMNNS	
Class	Partial Accur.	F-Mea.	Partial Accur.	F-Mea.	Partial Accur.	F-Mea.
Water	0.97	0.964	1	0.982	1	0.999
Marsh	0.854	0.784	0.97	0.843	0.965	0.884
Roads	0.767	0.825	0.936	0.948	0.955	0.956
Low. Veg.	0.759	0.788	0.792	0.845	0.856	0.872
Middle Veg.	0.502	0.509	0.705	0.777	0.698	0.782
High Veg.	0.629	0.601	0.805	0.796	0.957	0.914
Buildings	0.798	0.784	0.883	0.916	0.959	0.962
Landfills	0.335	0.459	0.182	0.308	0.364	0.533
Minimun	0.335	0.459	0.182	0.308	0.364	0.533
Mean	0.661	0.686	0.717	0.747	0.791	0.826

To perform the hold-out test, 7501 test instances (cells) were chosen in the Huelva study area, and 2320 were chosen in the Trabada study area which means that approximately 10% of the total number of instances of each dataset was used as test data. Table 4 and Table 6 show the results obtained by the SVM, SVMMRF, and SVMNNS methods, respectively. The tables show the number of test instances per class, the values for the errors of commission and omission, the global accuracy, and the Kappa index of agreement (KIA). In Table 5 and Table 7, the true positives and the F-measure per class are shown.

Based on the results obtained for every test instance counted as "hit" or "fail", we carried out a statistical analysis in order to determine if the classifiers behave statistically different. With a Cochran's Q test, we found that there exists

Table 6. Hold-out percentage results for Trabada

		SVM		SVMMRF		SVMNNS	
Class	Num. instan.	Comm. error	Omiss. error	Comm. error	Omiss. error	Comm. error	Omiss. error
Roads	640	38.00	23.40	35.50	17.90	18.00	12.20
Low. Veg.	549	12.60	32.40	4.20	31.00	4.60	18.00
High Veg.	765	14.10	21.10	5.50	16.70	7.30	10.70
Buildings	366	40.20	15.40	49.50	1.10	21.90	1.00
KIA		0.67		0.72		0.84	
Accuracy		75.56		79.61		88.10	

Table 7. Per-class partial accuracies for Trabada

	SVM		SVMMRF		SVMNNS	
Class	Partial Accur.	F-Mea.	Partial Accur.	F-Mea.	Partial Accur.	F-Mea.
Roads	0.62	0.686	0.645	0.723	0.82	0.848
Low. Veg.	0.874	0.763	0.958	0.802	0.954	0.882
High Veg.	0.859	0.822	0.945	0.885	0.927	0.91
Buildings	0.598	0.701	0.505	0.669	0.781	0.873
Minimun	0.598	0.686	0.505	0.669	0.781	0.848
Mean	0.738	0.743	0.763	0.770	0.870	0.878

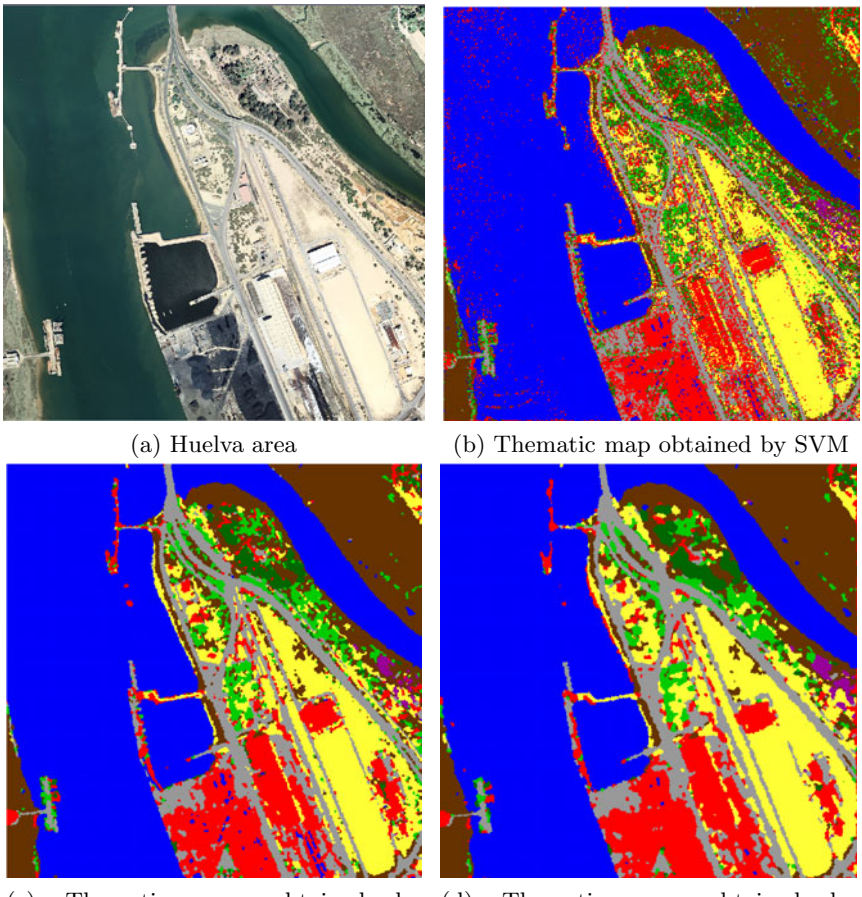
Table 8. Holm's adjusted p-values for the significance McNemar's tests

	algorithm	χ^2	p-value	Holm's
2	SVMMRF	300.62	1.32^{-10}	0.025
1	SVM	908.40	3.17^{-10}	0.05

a significant difference in usage among the three methods we surveyed ($p_{value} < 10^{-9}), \alpha = 0.05$). A posterior post-hoc analyis based on pairwise comparison using McNemar's tests (traditional non-parametric test in remote sensing [15]) with Holm's correction revealed that SVMNNS significantly obtained different accuracies for both areas as can be seen in Table 8.

Having found that the differences among the methods were significantly different for $\alpha = 0.05$, the analysis of the results concluded that the accuracy of the SVMNNS method was significantly better than that of its competitors for the data of the study areas.

Finally, the aerial images and the obtained thematic maps of Huelva and Trabada are shown in Figures 2 and 3 for visual comparison.



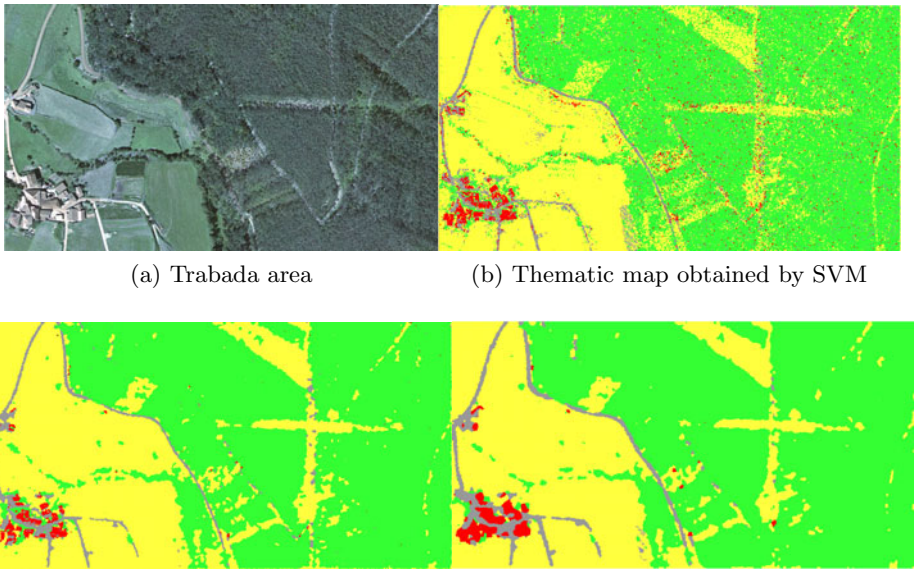
(a) Huelva area

(b) Thematic map obtained by SVM

(c) Thematic map obtained by SVMMRF

(d) Thematic map obtained by SVMNNS

Fig. 2. Final classification obtained for Huelva by each model. Water in blue, marshland in brown, roads and railways in grey, low vegetation and bare earth in yellow, middle vegetation in light green, eucalyptus in green, buildings in red and landfills in purple.



(a) Trabada area (b) Thematic map obtained by SVM

(c) Thematic map obtained by SVMMRF (d) Thematic map obtained by SVMNNS

Fig. 3. Final classification obtained for Trabada by each model. Roads in grey, low vegetation and bare earth in yellow, eucalyptus in green and buildings in red.

3.2 Discussion

The results for Huelva show a high level of accuracy (Table 4), with global accuracies above 81% and the contextual treatment improved the results in more than 8 points. Comparing both contextual methods, we can see SVMNNS obtained the best results for accuracy, both globally (almost three percentage points of difference from SVMMRF) and partially (see Table 5), except for the classes for medium vegetation and marshlands, categories for which SVMMRF performed better but with SVMNNS very close. If F-measure is selected as comparing measure in the place of partial accuracy, we will see that SVMNNS outperformed SVMMRF in every single category.

From a visual point of view (Figure 2, it can be observed that the contextual treatment improved the general results of SVM overcoming "salt and pepper" problems. An important issue related to this dataset is its clear problem of imbalance [3] (e.g., 2.151 instances of water and 209 of landfills). In any case, SVMNNS presents better behavior for the minority classes, in contrast to the SVMMRF and SVM methods (see F-measure for landfills in Table 5).

The Trabada area only had four classes, and although one could presume that the results would be better, the accuracies did not quite reach 80% except for SVMNNS (Table 6). This fact can be attributed to a lower level of separability from the available bands among classes and the influence of atypical instances that produce mistakes. The difference between SVMNNS and SVMMRF is much

higher in this case. As SVMNNS do not depend on the initial classification except for the first iteration, its results outperformed SVMMRF. However, it is important to stress that the use of the context notably improves the results of SVM (almost 4 points of difference in the worst case).

Visually (Figure 3), the final classification shows that SVMNNS is better than SVMMRF, which tends to introduce areas of eucalyptus in zones of buildings. Finally, it is worth noting the problems found with classification for every classifier when working on the road label. This may be due to the relatively similar infrared response that this class has compared with other classes.

By jointly observing the results, other conclusions can be made. First, it is worth underlining the synergy between LIDAR and orthophotography to separate the classes of both study areas. It is important to keep in mind that high levels of accuracy have been reached ($> 85\%$) for SVMMRF and SVMNNS with both datasets. In addition, SVMNNS improved the SVM and SVMMRF results not only globally but also in general for each label according to the partial accuracies. This improvement with respect to SVMMRF was greater for the Trabada data than for the Huelva data. That is, better performance is obtained with the greater resolution of evaluated data. The strategy of the classifier combination turned out to be more appropriate in this context than the parametric MRF approach to develop a contextual regularization, which, as has been observed, may generate a very high level of errors in certain cases. However, when the training set is easier to handle, the differences between the two contextual approaches are reduced, although SVMNNS maintains a higher level of accuracy for the results.

4 Conclusion

This study presented a novel method, called SVMNNS, to generate thematic maps based on data fusion with the objective of improving the return of investment for LIDAR and orthophotography joint flights. The results obtained by the SVMNNS method were compared with other classifiers (SVM and SVMMRF), and this comparison showed that SVMNNS obtained the best results in two distinct areas of the Iberian Peninsula.

Despite the good results that were obtained, there are still problems to be resolved. The most important issues are related to the method of extracting training data and their possible imbalances. The introduction of errors, the lack of accuracy, and the lack of completeness of the data provided by experts are problems that severely affect classification. To solve these problems, SVMNNS should evolve from supervised learning to active learning [16] introducing a detection phase for singular points, considering whether to eliminate them if they are possible outliers or to suggest their introduction within the training database if they are possible key instances. In addition, the application of new optimization techniques, such as evolutionary computation, may improve the final results by assigning weights to features, thus correcting problems associated with certain data sources e.g., those caused by intensities from multiple LIDAR returns.

References

1. Anderson, J., Plourde, L., Martin, M., Braswell, B., Smith, M., Dubayah, R., Hofton, M., Blair, B.: Integrating waveform lidar with hyperspectral imagery for inventory of a northern temperate forest. Remote Sensing of Environment 112(4), 1856–1870 (2008)
2. Brzank, A., Heipke, C., Goepfert, J., Soergel, U.: Aspects of generating precise digital terrain models in the Wadden Sea from lidar water classification and structure line extraction. ISPRS Journal of Photogrammetry and Remote Sensing 63, 510–528 (2008)
3. Chawla, N., Japkowicz, N., Kolcz, A.: Editorial: Special issue on learning from imbalanced data sets. ACM SIGKDD 6(1), 1–6 (2004)
4. Cortijo, F.J., de la Blanca, N.P.: Improving classical contextual classifications. International Journal of Remote Sensing 19(8) (1998)
5. Garcia, M., Riaño, D., Chuvieco, E., Danson, F.: 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)
6. Garcia-Gutierrez, J., Mateos-Garcia, D., Riquelme-Santos, J.C.: EVOR-STACK: a label-dependent evolutive stacking on remote sensing data fusion. Neurocomputing 75(1), 115–122 (2012)
7. Garcia-Gutierrez, J., Mateos-Garcia, D., Riquelme-Santos, J.C.: A SVM and k-NN Restricted Stacking to Improve Land Use and Land Cover Classification. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010, Part II. LNCS, vol. 6077, pp. 493–500. Springer, Heidelberg (2010)
8. García-Gutiérrez, J., Mateos-García, D., Riquelme-Santos, J.C.: Evor-stack: A label-dependent evolutive stacking on remote sensing data fusion. Neurocomputing 75(1), 115–122 (2012)
9. Goncalves-Seco, L., Miranda, D., Crecente, R., Farto, J.: Digital terrain model generation using airborne LIDAR in forested area of Galicia, Spain. In: Accuracy 2006, pp. 169–180 (2006)
10. 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)
11. Hofle, B., Pfeifer, N.: Correction of laser scanning intensity data: Data and model-driven approaches. ISPRS Journal of Photogrammetry and Remote Sensing 62(6), 415–433 (2007)
12. Huang, C.L., Wang, C.J.: A GA-based feature selection and parameters optimization for support vector machines. Expert Systems with Applications 31(2), 231–240 (2006)
13. Keerthi, S., Shevade, S., Bhattacharyya, C., Murthy, K.: Improvements to Platt's SMO algorithm for SVM classifier design. Neural Computation 13(3), 637–649 (2001)
14. Koetz, B., Morsdorf, F., van der Linden, S., Curt, T., Allgower, B.: Multi-source land cover classification for forest fire management based on imaging spectrometry and LiDAR data. Forest Ecology and Management 256, 263–271 (2008)
15. Tarabalka, Y., Fauvel, M., Chanussot, J., Benediktsson, J.: SVM- and MRF-based method for accurate classification of hyperspectral images. IEEE Geoscience and Remote Sensing Letters 7(4), 736–740 (2010)

16. Tuia, D., Pasolli, E., Emery, W.: Using active learning to adapt remote sensing image classifiers. Remote Sensing of Environment 115(9), 2232–2242 (2011)
17. Verrelst, J., Geerling, G., Sykora, K., Clevers, J.: Mapping of aggregated floodplain plant communities using image fusion of CASI and LiDAR data. International Journal of Applied Earth Observation and Geoinformation (11), 83–94 (2009)
18. Zhang, S., Zhang, C., Yang, Q.: Data preparation for data mining. Applied Artificial Intelligence 17(5/6), 375–381 (2003)

Spherical CIELab QAMs: Associative Memories Based on the CIELab System and Quantales for the Storage of Color Images

Marcos Eduardo Valle[1], Peter Sussner[2], and Estevão Esmi[2,⋆]

[1] Department of Mathematics, University of Londrina, Londrina - PR, Brazil
[2] Math. Imaging and Comp. Intelligence Group, University of Campinas, Campinas - SP, Brazil

Abstract. A *quantale* is the mathematical structure obtained by enriching a complete lattice with an associative binary operation which commutes with the supremum operation. We refer to an associative memory model that performs operations in a quantale as a *quantale-based associative memory* (QAM). Examples of QAMs include many lattice-based models such as *gray-scale morphological associative memories* and *implicative fuzzy associative memories*. Besides introducing auto-associative QAMs, this paper presents a QAM model for the storage and recall of color patterns. Specifically, novel QAM models, referred to as *spherical CIELab QAMs*, are defined in terms of the spherical coordinates of the CIELab system with an ordering scheme and a binary operation that yields a quantale. Computational experiments reveal that the spherical CIELab QAMs exhibit some tolerance with respect to impulsive noise.

Keywords: neural associative memory, morphological neural networks, quantales, color images, noise tolerance.

1 Introduction

The last few years have witnessed an increasing interest in *associative memories* (AMs) for the storage and retrieval of color images [12,16,17,19]. As far as we know, there are two straightforward approaches for adapting gray-scale models to color patterns.

The first approach relies on the RGB system in which a color element x is expressed in terms of red, green, and blue components, i.e., $x = [x_r, x_g, x_b]$ [1,4]. Hence, a color image can be decomposed into three gray-scale images which are usually stored in three separated gray-scale AMs. This technique has been employed by Zheng et al. for the storage of color images in a class of Cohen-Grossberg networks [19]. Notwithstanding, any gray-scale AM model can be extended to cope with color patterns using this approach. On the down side, the lack of interaction between the color channels may affect the noise tolerance of the color AM.

The second approach, which has been used by Vázquez and Sossa for the storage of true-color images [16,17], is based on the 24-bit representation of digital color images. Precisely, an integer q from 0 to $2^{24} - 1$ is assigned to the color value by means of the

⋆ This work was supported in part by Fundação Araucária/SETI and by CNPq under grant no. 309608/2009-0.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 467–478, 2012.

equation $q = 256^2 x_r + 256 x_b + x_g$, where x_r, x_b, x_g in $\{0, 1, \ldots, 255\}$ denote the red, green, and blue components of the color element in the digital 8-bit RGB system.

One negative aspect of this approach is that different color elements can be assigned to near integers. For example, the integers 255 and 256 are assigned respectively to the pure blue and visually black elements whose RGB representations are $[0, 0, 255]$ and $[0, 1, 0]$. This example indicates that the RGB model, which represents the most popular color space used conventionally to store, process, and display color images, is not suited to quantify the perceptual difference between images [7]. Indeed, the Euclidean distance between two color elements in the RGB system may not reflect the color difference perceived by the human eyes. For instance, at higher illumination, the eye is more sensitive if the color has not been saturated [1]. Therefore, the aforementioned approaches are not recommended in application areas such as multimedia, telecommunications, and printing industry, where the perceptual quality of the restored image is very important.

In contrast to the aforementioned approaches, *sparsely connected auto-associative morphological memories* (SCAMMs) do not rely on gray-scale models [12] since they can be defined on any complete lattice, a mathematical structure that is given by a partial ordering on a set. In particular, the set of colors in the CIELab system can be equipped with a partial ordering scheme that depends on the distance with respect to a certain color reference [14]. Since the CIELab is a perceptually uniform color space in the sense that the Euclidean distance between two color points corresponds to the perceptual difference by the human visual system [1], the CIELab-based SCAMMs can be used in applications that emphasize the perceptual image quality.

In comparison to the mathematically simple SCAMMs, classical *morphological associative memory* (MAM) models are defined in a richer mathematical structure that includes a group operation apart from the lattice operations ("meet" and "join") [8,10]. We believe that this is the reason why MAMs exhibit a better error correction capability than the computationally even less expensive SCAMMs in certain applications concerning the storage and recall of gray-scale images. These observations motivated us to investigate quantale-based auto-associative memories.

This paper is organized as follows: Section 2 discusses the mathematical background on the notion of a *quantale* that consists of a complete lattice with an associative binary operation. A class of AM models defined on quantales is introduced in Section 3. Section 4 introduces *spherical CIELab quantale-based auto-associative memories* and includes computational experiments concerning the storage and recall of large color images.

2 The Mathematical Framework: Quantales

The auto-associative memories introduced in this paper are defined in an algebraic structure called *(unital) quantale* [6,9]. A quantale is a complete lattice $\mathbb{Q}$ with an associative binary operation "$\cdot$", called multiplication, that distributes from both sides over arbitrary supremums. Precisely, recall that a partially ordered set $\mathbb{Q}$ is a complete lattice if every subset of $\mathbb{Q}$ has an infimum and a supremum in $\mathbb{Q}$ [2]. We denote the supremum and the infimum of a set $X \subseteq \mathbb{Q}$ by $\bigvee X$ and $\bigwedge X$, respectively. In particular,

$\bigvee_{j=1}^{n} x_j$ and $\bigwedge_{j=1}^{n} x_j$ are used to represent, respectively, the supremum (or maximum) and the infimum (or minimum) of a finite set $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{Q}$. Therefore, the operation "$\cdot$" is distributive over arbitrary supremums if the following equations hold true for every $q \in \mathbb{Q}$ and for every non-empty set $X \subseteq \mathbb{Q}$:

$$q \cdot \left(\bigvee X\right) = \bigvee_{x \in X} \{q \cdot x\} \quad \text{and} \quad \left(\bigvee X\right) \cdot q = \bigvee_{x \in X} \{x \cdot q\}. \tag{1}$$

It follows from (1) that the multiplication is increasing in both arguments, i.e., if $x \leq y$ then both inequalities $x \cdot q \leq y \cdot q$ and $q \cdot x \leq q \cdot y$ hold true for all $q \in \mathbb{Q}$ [9]. Moreover, the least element of $\mathbb{Q}$, denoted by the symbol "$\bot$", is a zero or absorbing element of the quantale $\mathbb{Q}$, i.e., $q \cdot \bot = \bot \cdot q = \bot$ for all $q \in \mathbb{Q}$.

We speak of a *commutative quantale* if the multiplication is commutative. Similarly, we speak of a *unital quantale* if the multiplication has an identity, i.e., if there exists $e \in \mathbb{Q}$ such that $e \cdot x = x \cdot e = x$ for all $x \in \mathbb{Q}$.

The binary operation "$\cdot$" of a quantale $\mathbb{Q}$ is always residuated [9]. Thus, there is a binary operation "$/$" in $\mathbb{Q}$ such that the following relationship holds for all $a, x, y \in \mathbb{Q}$:

$$a \cdot x \leq y \quad \Leftrightarrow \quad a \leq y/x. \tag{2}$$

The operation "$/$"" is called the *residual*, or *division*, of "$\cdot$". Furthermore, for any $x, y \in \mathbb{Q}$, this operation is uniquely determined by

$$y/x = \bigvee \{z \in \mathbb{Q} : z \cdot x \leq y\}. \tag{3}$$

In Section 4, we enrich the CIELab system with an ordering scheme and a binary operation that yields a quantale. The novel quantale is based on the following examples.

Example 1. The extended nonnegative real numbers $\mathbb{R}_{+\infty}^{\geq 0} = [0, +\infty]$, endowed with the usual "greater than or equal" relation, represents a complete lattice. Precisely, let the symbol "$\leq'$" denote the dual ordering given by $x \leq' y$ if and only if $y \leq x$. The algebraic structure $(\mathbb{R}_{+\infty}^{\geq 0}, \leq')$ is a complete lattice with largest element $0 = \bigvee' \mathbb{R}_{+\infty}^{\geq 0}$ and least element $+\infty = \bigwedge' \mathbb{R}_{+\infty}^{\geq 0}$. Furthermore, a commutative unital quantale arises by adding a binary operation $\times'$, which coincides with the usual multiplication on real numbers but satisfies the equations $(+\infty) \times' x = x \times' (+\infty) = +\infty$ for all x. The identity of the multiplication is also $e = 1$ but the zero is $+\infty$. The residual of the multiplication is the following binary operation

$$y/'x = \bigvee' \left\{z \in \mathbb{R}_{+\infty}^{\geq 0} : x \times' z \leq' y\right\} = \begin{cases} 0, & \text{if } x = +\infty \text{ or if } x = y = 0, \\ +\infty, & \text{if } x = 0, y > 0, \\ y \div x, & \text{otherwise,} \end{cases} \tag{4}$$

which coincides with the usual division with the exceptions $0/'0 = 0$ and $y/'(+\infty) = 0$ for all $y \in \mathbb{R}_{+\infty}^{\geq 0}$.

Example 2. Let $\Theta = (-\pi, \pi]$ and define the following relation for any $x, y \in \Theta$:

$$x \preceq y \quad \Leftrightarrow \quad \begin{cases} |x| < |y|, \text{ or} \\ |x| = |y| \text{ and } x \leq y. \end{cases} \tag{5}$$

The expression $x \prec y$ should be read as $x \preceq y$ but $x \neq y$. Note that $x \prec y$ if and only if we have that x is closer to 0 than y or that $|x| = |y|$ and $x < 0 < y$.

The algebraic structure $(\Theta, \preceq)$ constitutes a complete lattice with 0 and π as the least and largest elements. The minimum operation, denoted by the symbol "$\curlywedge$", is defined as follows for every $x, y \in \Theta$:

$$x \curlywedge y = \begin{cases} x, & \text{if } x \preceq y, \\ y, & \text{otherwise.} \end{cases} \tag{6}$$

Note that $(\Theta, \curlywedge)$ represents a unital commutative quantale. The identity of the multiplication "$\curlywedge$" is the largest element π. The division is the binary operation defined as follows for all $x, y \in \Theta$:

$$y^{\Theta}/x = \bigvee \{z \in \Theta : x \curlywedge z \preceq y\} = \begin{cases} \pi, & \text{if } x \preceq y, \\ y, & \text{otherwise.} \end{cases} \tag{7}$$

It is important to realize that any complete sublattice of a quantale is also a quantale. For instance, the set $\Phi = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, which is a complete sublattice of $(\Theta, \preceq)$, is also a commutative unital quantale with the binary operation "$\curlywedge$". In this case, however, the division is the binary operation defined as follows for all $x, y \in \Phi$:

$$y^{\Phi}/x = \begin{cases} \frac{\pi}{2}, & \text{if } x \preceq y, \\ y, & \text{otherwise.} \end{cases} \tag{8}$$

3 The Quantale-Based Auto-associative Memories

The single-layer quantale-based auto-associative memories (QAMs) considered in this paper are similar to the recursive memories investigated by Hopfield, Amari, Anderson, and others [5], but these models are defined in a quantale $(\mathbb{Q}, \cdot)$ instead of the usual ring $(\mathbb{R}, +, \times)$.

Suppose that the set that comprises the fundamental memories is the Cartesian product of a commutative unital quantale $\mathbb{Q}$, i.e., $\mathcal{X} = \mathbb{Q}^n$, and let $H = (h_{ij}) \in \mathbb{Q}^{n \times n}$ denote the synaptic weight matrix of the memory. Since the number of entries of the synaptic weight matrix $H \in \mathbb{Q}^{n \times n}$ of a fully connected network may grow impractically large as the dimension n increases, we focus on sparsely connected networks in this paper. Formally, let $\mathcal{T} \subseteq \mathcal{N} \times \mathcal{N}$, where $\mathcal{N} = \{1, \ldots, n\}$, denote a network topology in the sense that the i-th output depends directly on the j-th input if $(i, j) \in \mathcal{T}$. In this case, we only store the synaptic weights h_{ij} such that $(i, j) \in \mathcal{T}$. A fully connected network is obtained by considering $\mathcal{T} = \mathcal{N} \times \mathcal{N}$.

Given an input pattern $\mathbf{x} = [x_1, \ldots, x_n]^T \in \mathbb{Q}^n$, we set $\mathbf{x}(0) = \mathbf{x}$ and define the t-step QAM model, for $t = 1, 2, \ldots$, as the mapping $\mathcal{H}_t : \mathcal{X} \to \mathcal{X}$ which yields the t-th term of the following recursive sequence:

$$x_i(t) = \bigvee_{(i,j) \in \mathcal{T}} \{h_{ij} \cdot x_j(t-1)\}, \quad \forall\, i = 1, 2, \ldots, n. \tag{9}$$

If the sequence $\{\mathbf{x}(t)\}_{t=0}^{\infty}$ is convergent for any input pattern $\mathbf{x} \in \mathbb{Q}^n$, then we also define the $*$-QAM model as the mapping $\mathcal{H}_* : \mathcal{X} \to \mathcal{X}$ given by $\mathcal{H}_*(\mathbf{x}) = \lim \mathbf{x}(t)$.

Note that $\mathcal{H}_t(\mathbf{x})$ represents the output of a synchronous dynamic model after t steps. In particular, $\mathcal{H}_1(\mathbf{x})$ corresponds to the output of a single step model. An asynchronous model can be defined in a similar fashion by updating only one neuron at each step.

Given a fundamental memory set $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^p\} \subseteq \mathbb{Q}^n$ and a certain network topology $\mathcal{T} \subseteq \mathcal{N} \times \mathcal{N}$, the sparsely connected synaptic weight matrix $H \in \mathbb{Q}^{n \times n}$ of a QAM model given by (9) can be defined as follows:

$$H = \bigvee \left\{ A \in \mathbb{Q}^{n \times n} : \bigvee_{(i,j) \in \mathcal{T}} \{a_{ij} \cdot x_j^\xi\} \leq x_i^\xi, \ \forall i \in \mathcal{N}, \forall \xi = 1, \ldots, p \right\}. \tag{10}$$

Alternatively, the relevant entries of the synaptic weight matrix H given by (10) can be easily computed by means of the following equation:

$$h_{ij} = \bigwedge_{\xi=1}^{p} \left(x_i^\xi / x_j^\xi \right), \quad \forall \, (i,j) \in \mathcal{T}. \tag{11}$$

Since the entries of h_{ij} are determined using the right residual "/" of the multiplication of the quantale $\mathbb{Q}$, we refer to (10) as the *residual recording recipe*. The recording recipe given by (10) is also refered to as *adjunction-based learning* in view of the adjunction relationship between the operations "$\cdot$" and "/" expressed by (2) [15].

The residual recording recipe is optimal in the following sense: If there exists $A \in \mathbb{Q}^{n \times n}$ such that $x_i^\xi = \bigvee_{(i,j) \in \mathcal{T}} \{a_{ij} \cdot x_j^\xi\}$, then (10) yields $H \in \mathbb{Q}^{n \times n}$ such that $x_i^\xi = \bigvee_{(i,j) \in \mathcal{T}} \{h_{ij} \cdot x_j^\xi\}$ and $a_{ij} \leq h_{ij}$ for all $(i,j) \in \mathcal{T}$. Also note that the sequence $\{\mathbf{x}(t)\}_{t=0}^{\infty}$ given by (9) is well defined and convergent if the synaptic weight matrix H is given by the residual recording recipe with a network topology that allows self-connections, i.e., if $(i,i) \in \mathcal{T}$ for all $i \in \mathcal{N}$. Moreover, as the following theorem reveals, this sequence is closely related to the set

$$\mathcal{F} = \left\{ \mathbf{z} = [z_1, z_2, \ldots, z_n]^T \in \mathbb{Q}^n : z_i = \bigvee_{(i,j) \in \mathcal{T}} \{h_{ij} \cdot z_j\}, \ \forall i \in \mathcal{N} \right\}, \tag{12}$$

of all fixed points of the synaptic weight H subject to $\mathcal{T}$. The following theorem also shows that, independently of the number of items, the fundamental memories $\mathbf{x}^1, \ldots, \mathbf{x}^p$ belong to $\mathcal{F}$. As a consequence, the QAM model exhibits optimal absolute storage capacity.

Theorem 1. *Given a fundamental memory set $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^p\} \subseteq \mathbb{Q}$, define H by means of (10) with a network topology $\mathcal{T}$ that allows self-connections. Then, the following relations hold true for any input $\mathbf{x} \in \mathbb{Q}^n$ and for all $t \geq 1$:*

$$\mathbf{x} \leq \mathcal{H}_t(\mathbf{x}) \leq \mathcal{H}_{t+1}(\mathbf{x}) \leq \mathcal{H}_*(\mathbf{x}) = \bigwedge \{\mathbf{z} \in \mathcal{F} : \mathbf{z} \geq \mathbf{x}\}. \tag{13}$$

Furthermore, $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^p\} \subseteq \mathcal{F}$ and, therefore, the equation $\mathcal{H}_t(\mathbf{x}^\xi) = \mathbf{x}^\xi$ holds for all $\xi = 1, \ldots, p$ and for any integer t.

Gray-scale MAMs, that are defined on the quantales $(\mathbb{Q}, +)$ or $(\mathbb{Q}, +')$, where $\mathbb{Q}$ equals $\mathbb{R}_{\pm\infty}$ or $\mathbb{Z}_{\pm\infty}$, are examples of QAMs with the residual recording recipe [10]. *Implicative fuzzy associative memories* (IFAMs), defined on the quantale consisting of the unit interval $[0, 1]$ with a left-continuous triangular norm, also constitute QAMs with residual recording recipe [11]. Since both gray-scale MAMs and IFAMs are fully connected networks, i.e., $\mathcal{T} = \mathcal{N} \times \mathcal{N}$, they satisfy Theorem 1.

4 The Spherical CIELab QAM

Let us begin this section by recalling that a color image $\mathbf{x}$ is a function from a spatial domain $\mathcal{D}$ into a set of color values. For simplicity, we consider only finite spatial domains, i.e., $\mathcal{D} = \{d_1, \ldots, d_n\}$. Moreover, we suppose that the set of color values corresponds to the RGB space, denoted by $\mathbb{V}_{RGB}$. The RGB color model is based on the *tristimulus theory* of vision which states that colors are perceived by our visual system as a combination of the three primary colors: *red* (R), *green* (G), and *blue* (B) [1,4]. Thus, a color x in the RGB model corresponds to a point $[x_r, x_g, x_b]$ on or inside the cube $\mathbb{V}_{RGB} = [0, 1] \times [0, 1] \times [0, 1]$. An RGB color image $\mathbf{x}$ corresponds to an array $[x_1, x_2, \ldots, x_n] \in \mathbb{V}_{RGB}^n$, where $x_j = \mathbf{x}(d_j)$ for every $j = 1, \ldots, n$. For example, Figure 1 shows twelve RGB color images of size 384×256 pixels. Each of these images corresponds to a pattern $\mathbf{x}^\xi \in \mathbb{V}_{RGB}^n$, where $n = 98304$.

The RGB model is widely used in hardware devices including image scanners, digital cameras, and liquid-crystal display televisions. Hence we assume that noise is introduced in a digital color image using this color system. For example, an RGB color image may be corrupted by additive Gaussian noise due to faulty sensors or impulsive noise introduced by environmental interference or faulty communication [7]. An image corrupted by Gaussian noise is obtained by adding a term drawn from a zero mean normal distribution with variance σ^2 at each RGB component of all picture elements. In contrast, an impulsive noise is modeled as follows for all $i = 1, \ldots, n$. The symbols $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbb{V}_{RGB}^n$ and $\tilde{\mathbf{x}} = [\tilde{x}_1, \ldots, \tilde{x}_n] \in \mathbb{V}_{RGB}^n$ denote respectively the original and the corrupted image.

$$[(\tilde{x}_r)_i, (\tilde{x}_g)_i, (\tilde{x}_b)_i] = \begin{cases} [(x_r)_i, (x_g)_i, (x_b)_i], & \text{with probability } 1 - p, \\ [d, (x_g)_i, (x_b)_i], & \text{with probability } p_r p, \\ [(x_r)_i, d, (x_b)_i], & \text{with probability } p_g p, \\ [(x_r)_i, (x_g)_i, d], & \text{with probability } p_b p, \\ [d, d, d], & \text{with probability } p_s p. \end{cases} \tag{14}$$

Here, p is the probability that a picture element is corrupted by noise. The terms p_r, p_g, and p_b, satisfying $p_r + p_g + p_b \leq 1$, represent the probabilities of noise in the red, green, and blue channels, respectively, and $p_s = 1 - p_r - p_g - p_b$ corresponds to the probability of simultaneous noise in all channels. Finally, $d \in \{0, 1\}$ is the impulsive value drawn from a Bernoulli distribution with equal probability of success and failure. The first column of Figure 2 displays the image "parrots" corrupted by impulsive noise with probabilities $p = 0.1$, $p_r = p_g = p_b = 0.25$ and the image "caps" corrupted by Gaussian noise with variance 0.01.

Fig. 1. Original color images $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{12}$ of size 384×256 pixels

In many application areas such as multimedia, telecommunications, and printing industry, the emphasis is on the perceptual quality of the restored image [7]. Despite its popularity as a color space for storing, processing, and displaying color images, the RGB model is not an appropriate color space to quantify the perceptual difference between images. Thus, we defined the QAM model in a variation of the CIELab model, which is a perceptually uniform color space in the sense that the Euclidean distance between two color points approximates the perceptual difference between the two colors by the human vision system.

4.1 The Spherical CIELab Quantale

The CIELab color system, which was adopted as an international standard in the 1970s by the *International Commission on Illumination* (CIE - Commission Internationale de l'Eclairage), encompasses the entire visible spectrum and can accurately represent the colors of many input and output devices, including monitors, printers, digital cameras, etc. [1,4]. For simplicity, let us suppose that the CIELab color space corresponds to the set $\mathbb{V}_{Lab} = [0, 100] \times \mathbb{R} \times \mathbb{R}$. In practice, however, only a finite subset of $\mathbb{V}_{Lab}$ is used

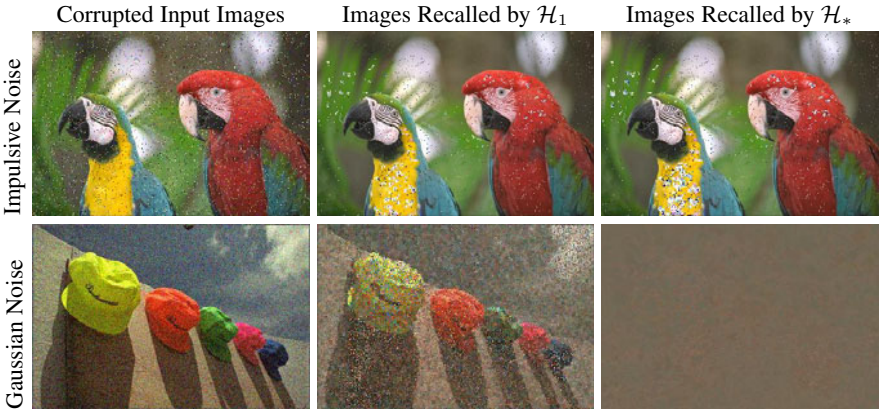


Fig. 2. The first column shows the images $\tilde{\mathbf{x}}^1$ and $\tilde{\mathbf{x}}^2$ corrupted, respectively, by impulsive and Gaussian noise. The following columns depict the corresponding images recalled by $\mathcal{H}_1$ and $\mathcal{H}_*$.

in view of the finite representation of digital color images. Let $\zeta : \mathbb{V}_{RGB} \to \mathbb{V}_{Lab}$ and $\zeta^\dagger : \mathbb{V}_{Lab} \to \mathbb{V}_{RGB}$ denote the non-linear mappings that perform the conversion between the RGB and CIELab systems. Codes implementing these two mappings have been uploaded by Mark Ruzon at the *file exchange* section of `www.mathworks.com`.

The following interpretation are attributed to the coordinates u_L, u_a, and u_b of a point $u \in \mathbb{V}_{Lab}$ [4]:

1. The coordinate u_a encodes the red-green sensation. More precisely, positive values indicate red colors while negative values indicate green colors.
2. The coordinate u_b refers to the yellow-blue sensation where positive and negative values correspond to yellow and blue colors, respectively.
3. Finally, the first coordinate u_L models the lightness. The locations $(0, 0, 0)$ and $(100, 0, 0)$ correspond respectively to the black and white colors. Gray-scale or achromatic points are located on the line segment determined by $u_a = 0$, $u_b = 0$, and u_L between 0 and 100.

In addition, the value $\sqrt{u_a^2 + u_b^2}$ represents the *chroma*, which measures the colorfulness of $u \in \mathbb{V}_{Lab}$ compared to white.

Since CIELab is a perceptually uniform color space, we choose to rank the colors according to the Euclidean distance from a certain reference element $r = [r_L, r_a, r_b] \in \mathbb{V}_{Lab}$. This remark can be easily accomplished by using spherical coordinates. Specifically, let $\mathbb{S} = \mathbb{R}^{\geq 0} \times \Phi \times \Theta$ denote the spherical CIELab system centered at r (cf. Examples 1 and 2) and let $\varsigma : \mathbb{V}_{Lab} \to \mathbb{S}$ and $\varsigma^\dagger : \mathbb{S} \to \mathbb{V}_{Lab}$ be the mappings (correspond to the `MATLAB` routines `cart2sph` and `sph2cart`) between $\mathbb{V}_{Lab}$ and $\mathbb{S}$.

Given two points $u = [u_\rho, u_\phi, u_\theta] \in \mathbb{S}$ and $v = [v_\rho, v_\phi, v_\theta] \in \mathbb{S}$, let us define

$$u \preceq_{\mathbb{S}} v \;\Leftrightarrow\; \begin{cases} v_\rho < u_\rho, \text{ or} \\ v_\rho = u_\rho \text{ and } u_\phi \prec v_\phi, \text{ or} \\ v_\rho = u_\rho, u_\phi = v_\phi, \text{ and } u_\theta \preceq v_\theta, \end{cases} \tag{15}$$

where $x \prec y$ is given by Equation 5 and the subsequent comment in Example 2.

In general terms, the lexicographical ordering given by (15) expresses that the color v is closer to the reference point r than the color u. Indeed, if $u \leq_{\mathbb{S}} v$, then v is in or inside the sphere that is centered at r and that contains u. The second and third inequalities in (15) avoid ambiguities by comparing the angles between the two colors. In particular, if the second condition in (15) is satisfied, then the chroma of v is closer to the chroma of the reference point r than that of u. Also, observe that the last conditions in (15) makes $\leq_{\mathbb{S}}$ a total ordering scheme.

The largest element of $(\mathbb{S}, \leq_{\mathbb{S}})$ is $[0, 0, 0]$, which corresponds to the reference point r in the CIELab system $\mathbb{V}_{Lab}$. However, $\mathbb{S}$ is not a complete lattice because it does not have a least element. We can overcome this problem by adding an artificial point $\perp = [+\infty, 0, 0]$ representing the least element. Thus, the set $\bar{\mathbb{S}} = \mathbb{S} \cup \{\perp\}$ is a complete lattice with respect to the total ordering $\leq_{\mathbb{S}}$ given by (15).

We are now able to endow $\bar{\mathbb{S}}$ with a binary operation $\otimes$ such that $(\bar{\mathbb{S}}, \otimes)$ constitutes a commutative unital quantale. Given two color elements $u, v \in \bar{\mathbb{S}}$, we define

$$u \otimes v = [u_\rho \times' v_\rho, u_\phi \curlywedge v_\phi, u_\theta \curlywedge v_\theta], \tag{16}$$

where $\times'$ and $\curlywedge$ are the binary operations defined in Examples 1 and 2.

Note that the operation $\otimes$, which is defined in a component-wise manner, inherits the associativity and commutativity of $\times'$ and $\curlywedge$. Furthermore, since $\times'$ and $\curlywedge$ distribute respectively over supremums in the complete lattice $(\mathbb{R}^{\geq 0}_{+\infty}, \leq')$ and any complete sub-lattice of $(\Theta, \preceq)$, we conclude that the operation $\otimes$ distributes over supremums in $(\bar{\mathbb{S}}, \leq_{\mathbb{S}})$. Also, note that $e = (1, \frac{\pi}{2}, \pi) \in \bar{\mathbb{S}}$ is the identity of $\otimes$, i.e.,

$$e \otimes v = [1 \times' v_\rho, \left(\frac{\pi}{2}\right) \curlywedge v_\phi, \pi \curlywedge v_\theta] = [v_\rho, v_\phi, v_\theta] = v, \quad \forall v \in \bar{\mathbb{S}}. \tag{17}$$

The right division of the multiplication $\otimes$ is the binary operation defined as follows for any color elements $u, v \in \bar{\mathbb{S}}$:

$$v \oslash u = \left(v_\rho /' u_\rho, v_\phi \,{}^{\Phi}\!/ u_\phi, v_\theta \,{}^{\Theta}\!/ u_\theta\right), \tag{18}$$

where "$/'$", "${}^{\Phi}\!/$", and "${}^{\Theta}\!/$" denote the divisions given respectively by (4), (8), and (7).

4.2 The Spherical CIELab QAM for the Storage and Recall of RGB Patterns

A class of QAM models for the storage and recall of RGB patterns can be defined by taking advantage of the conversion mappings $\varsigma, \varsigma^\dagger, \zeta$, and $\zeta^\dagger$ as depicted in Figure 3. Specifically, let boldfaced letters denote the component-wise application of each one of the four conversion mappings. Suppose that we want to store RGB patterns $\mathbf{x}^\xi \in \mathbb{V}^n_{RGB}$ in an associative memory model based on the quantale $(\bar{\mathbb{S}}, \otimes)$. An application of $\boldsymbol{\varsigma\zeta}$ converts $\mathbf{x}^\xi$ into a pattern $\mathbf{u}^\xi \in \bar{\mathbb{S}}^n$ that is stored in the QAM using the residual recording recipe with a certain network topology $\mathcal{T} \in \mathcal{N} \times \mathcal{N}$. Afterwards, given an RGB pattern $\mathbf{x} \in \mathbb{V}^n_{RGB}$, we compute the spherical CIELab pattern $\mathbf{u} = \boldsymbol{\varsigma\zeta}(\mathbf{x})$ and feed it as an input into the QAM. The resulting output $\mathbf{v} \in \bar{\mathbb{S}}$ is then transformed into

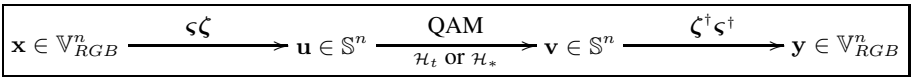


Fig. 3. Strategy for the storage and recall of RGB patterns using the spherical CIELab QAM

an RGB pattern $\mathbf{y} \in \mathbb{V}^n_{RGB}$ by applying respectively the mappings $\varsigma^\dagger$ and $\zeta^\dagger$, i.e., $\mathbf{y} = \zeta^\dagger\varsigma^\dagger(\mathbf{v})$.

It is important to note that the QAM output $\mathbf{v}$ is never equal to the least element $\perp = [\perp, \perp, \ldots, \perp]^T$ of $\bar{\mathbb{S}}^n$. Indeed, $\mathbf{x} \in \mathbb{V}^n_{RGB}$ implies $\perp \neq \mathbf{u} = \varsigma\zeta(\mathbf{x})$ and, by Equation (13), we conclude that $\perp < \mathbf{u} \leq \mathcal{H}_t(\mathbf{u}) \leq \mathcal{H}_*(\mathbf{u})$ for all $t \geq 1$. Also, in view of the ordering given by (15), the recalled pattern is, at least, as close to the constant pattern $\mathbf{r} = [r, r, \ldots, r]^T$ as the input $\mathbf{u}$.

As an illustrative example, consider the RGB color images $\mathbf{x}^1, \ldots, \mathbf{x}^{12}$ depicted in Figure 1. First, we converted these twelve images into $\mathbf{u}^\xi \in \bar{\mathbb{S}}^n$, $n = 98304$, by applying respectively the mappings $\boldsymbol{\zeta}$ and $\boldsymbol{\varsigma}$. In this example, we used the mean of all color elements of all patterns as the reference CIELab color, i.e., we defined

$$r = \frac{1}{12n} \sum_{\xi=1}^{12} \sum_{i=1}^{n} \zeta(x_i^\xi). \tag{19}$$

Then, we stored the resulting patterns $\mathbf{u}^1, \ldots, \mathbf{u}^{12}$ in the spherical CIELab QAM using the residual recording recipe with a small world network topology with self-connections [18]. Specifically, our network topology was obtained by first connecting each node with itself and its 24 neighbors. Then a fraction of the edges of the neighbors have been rewired at random with probability 0.3. We would like to recall that, apart from associative memories models [3,13], small-world topologies can be found in many large, sparse networks in nature such as the neural network of the worm *Caenorhabditis Elegans*, the power grid of the western United States, and the collaborative graph of film actors [18].

We confirmed perfect recall of all undistorted patterns according to Theorem 1. Then we probed the QAM models with the corrupted images displayed in first column of Figure 2. The outcome of this experiment is visualized in the second and third columns of Figure 2. Specifically, the second column displays the images recalled by a single-step QAM $\mathcal{H}_1$ while the third column shows the images recalled by $\mathcal{H}_*$. The recursive $\mathcal{H}_*$ model converged within five steps.

Note that both $\mathcal{H}_1$ and $\mathcal{H}_*$ removed some of the impulsive noise from the corrupted image "parrots". In contrast, the two QAM models are not suited for the removal of Gaussian noise. Indeed, the recalled image "caps" almost converted to the corresponding RGB constant image of $\mathbf{r} = [r, r, \ldots, r]^T$.

Finally, we compared the impulsive noise removal capabilities of the novel QAMs and our previous *sparsely connected morphological associative memories* (SCAMMs) $\mathcal{W}^L_{RGB}$ and $\mathcal{W}^r_{Lab}$ in terms of the error measure ΔE^*_{ab} [12,14]. Recall that ΔE^*_{ab}, given by the following equation where $\|\cdot\|_2$ denotes the usual Euclidean norm, measures the distance between two images by taking into account the human color perception.

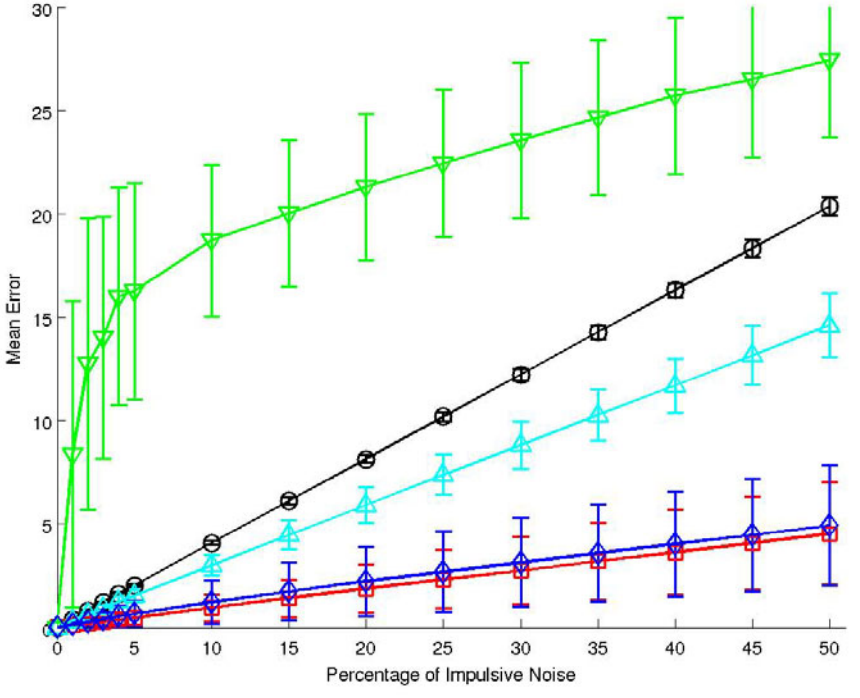


Fig. 4. Arithmetic mean of the distances ΔE^*_{ab} (with error bars symbolizing the standard deviations) between the original images $\mathbf{x}^\xi$ and the corrupted images $\tilde{\mathbf{x}}^\xi$ (marked with $\circ$) as well as the ones between $\mathbf{x}^\xi$ and the outputs for $\tilde{\mathbf{x}}^\xi$ produced the SCAMMs $\mathcal{W}^L_{RGB}$ (marked with $\bigtriangledown$) and $\mathcal{W}^r_{Lab}$ (marked with $\bigtriangleup$), and the QAMs $\mathcal{H}_1$ (marked with $\Diamond$) and $\mathcal{H}_*$ (marked with $\square$) in 120 simulations for each percentage of impulsive noise.

$$\Delta E^*_{ab}(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{j=1}^{n} \|\zeta(x_j) - \zeta(y_j)\|_2, \quad \text{for any } \mathbf{x}, \mathbf{y} \in \mathbb{V}^n_{RGB}. \tag{20}$$

Figure 4 shows the arithmetic means of the distances ΔE^*_{ab} between the original images and corrupted versions in 120 simulations, where each original image of Figure 1 was corrupted 10 times for each percentage of impulsive noise. This figure also shows the confidence intervals above and below the means as error bars. For a better comparison, we also included the distances between the original and the corrupted images.

Note that the QAM models outperformed our previous SCAMMs. Also, note that the QAM $\mathcal{H}_1$ produced slightly smaller errors than the $\mathcal{H}_*$, which converged in an average of 4.7 steps. We believe that these are promising results on the novel QAM models. In the future, we plan to perform further research on the noise tolerance of QAMs. We also intend to compare the performance of the novel models with other color models such as the Cohen-Grossberg memory introduced by Zheng et al. [19].

References

1. Acharya, T., Ray, A.: Image Processing: Principles and Applications. John Wiley and Sons, Hoboken (2005)
2. Birkhoff, G.: Lattice Theory, 3rd edn. American Mathematical Society, Providence (1993)
3. Bohland, J., Minai, A.: Efficient associative memory using small-world architecture. Neurocomputing, 38–40 (2001)
4. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice-Hall, Upper Saddle River (2002)
5. Hassoun, M.H. (ed.): Associative Neural Memories: Theory and Implementation. Oxford University Press, Oxford (1993)
6. Mulvey, C.J.: Rend. Circ. Mat. Palermo 12, 99–104 (1986)
7. Plataniotis, K., Androutsos, D., Venetsanopoulos, A.: Adaptive fuzzy systems for multichannel signal processing. Proceedings of the IEEE 87(9), 1601–1622 (1999)
8. Ritter, G.X., Sussner, P., de Leon, J.L.D.: Morphological associative memories. IEEE Transactions on Neural Networks 9(2), 281–293 (1998)
9. Russo, C.: Quantale modules and their operators, with applications. Journal of Logic and Computation 20(4), 917–946 (2010)
10. Sussner, P., Valle, M.E.: Grayscale morphological associative memories. IEEE Transactions on Neural Networks 17(3), 559–570 (2006)
11. Sussner, P., Valle, M.E.: Implicative fuzzy associative memories. IEEE Transactions on Fuzzy Systems 14(6), 793–807 (2006)
12. Valle, M.E.: A class of sparsely connected autoassociative morphological memories for large color images. IEEE Transactions on Neural Networks 20(6), 1045–1050 (2009)
13. Valle, M.E.: Sparsely connected autoassociative fuzzy implicative memories and their application for the reconstruction of large gray-scale images. Neurocomputing 74(1-3), 343–353 (2010)
14. Valle, M.E., Grande Vicente, D.M.: Some experimental results on sparsely connected autoassociative morphological memories for the reconstruction of color images corrupted by either impulsive or gaussian noise. In: Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2011), San Jose, CA, USA, pp. 275–282 (August 2011)
15. Valle, M.E., Sussner, P.: Storage and recall capabilities of fuzzy morphological associative memories with adjunction-based learning. Neural Networks 24(1), 75–90 (2011)
16. Vazquez, R.A., Sossa, H.: A bidirectional hetero-associative memory for true-color patterns. Neural Processing Letters 28(3), 131–153 (2008)
17. Vazquez, R.A., Sossa, H.: Behavior of morphological associative memories with true-color image patterns. Neurocomputing 73(1-3), 225–244 (2009)
18. Watts, D., Strogatz, S.: Collective dynamics of 'small-world' networks. Nature 393(6684), 440–442 (1998)
19. Zheng, P., Zhang, J., Tang, W.: Color image associative memory on a class of Cohen–Grossberg networks. Pattern Recognition 43(10), 3255–3260 (2010)

Fuzzy Associative Memories Based on Subsethood and Similarity Measures with Applications to Speaker Identification

Estevão Esmi[1,⋆], Peter Sussner[1,⋆], Marcos Eduardo Valle[2,⋆], Fábio Sakuray[3], and Laécio Barros[1,⋆]

[1] Department of Applied Mathematics, University Campinas, Campinas, São Paulo, Brazil
[2] Department of Mathematics, University of Londrina, Londrina, Paraná, Brazil
[3] Department of Computer Science, University of Londrina, Londrina, Paraná, Brazil

Abstract. Recently, we presented a non-distributive fuzzy associative memory (FAM) called the Kosko subsethood FAM, for short KS-FAM. This model can be classified as a morphological neural network because it is based on computing the degree of fuzzy inclusion or subsethood of patterns and this operation can be considered an erosion in fuzzy mathematical morphology. In this paper, we introduce a whole range of extensions of the KS-FAM called S-FAMs, dual S-FAMs, and SM-FAMs. Here, the acronyms S-FAM and SM-FAM stand for respectively subsethood FAM and similarity measure FAM. The new models share some properties with the KS-FAM such as unlimited absolute storage capacity and a small number of spurious memories. The paper finishes some experimental results concerning the problem of text-independent speaker identification. For comparative purposes, we included the recognition rates obtained by some well-known classifiers from the literature.

Keywords: fuzzy associative memory, subsethood, inclusion, similarity measure, speaker identification.

1 Introduction

Measures of subsethood and inclusion play a very important role in diverse areas such as fuzzy sets and systems, artificial neural networks, machine learning, and mathematical morphology [10,13,15,16,20,21,27,29,37]. Although the technical terms "fuzzy subsethood" and "fuzzy inclusion" are often used interchangeably, a fuzzy inclusion measure should – in contrast to a subsethood measure – satisfy the so called heritage property [17]. In other words, computing the fuzzy inclusion of a crisp set A in a crisp set B should yield 1 if $A \subseteq B$ and 0 if $A \not\subseteq B$.

Other than that, there is no general agreement among researchers what a fuzzy subsethood or inclusion measure should represent. Fuzzy subsethood and inclusion measures have been defined axiomatically and constructively [1,6,10,22,26,37,39]. In particular, fuzzy inclusion measures can be constructed in terms of infima of implications. In this

⋆ This work was supported by FAPESP under grants nos. $2009/16284-2$ and $2011/10014-3$, by CNPq under grants nos. $309608/2009-0$ and $306872/2009-9$, and by Fundação Araucária/SETI under grant no. $14-1-15.197$.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 479–490, 2012.

case, this definition gives rise to an erosion of a fuzzy image A by a structuring element S in fuzzy mathematical morphology (FMM) [7,8,23,30]. In accordance with the objective of an erosion in image processing, the degree of inclusion S_x, i.e. the structuring element S centered at x, in the image A is computed at every pixel location x. Moreover, since this type of operation commutes with the infimum operator for an arbitrary, but fixed structuring element S, it represents an algebraic erosion in the lattice-algebraic sense. Dual operators called fuzzy dilations can be constructed in terms of suprema of fuzzy conjunctions. Pairs consisting of a fuzzy erosion and dilation form the basis for a certain approach towards FMM whose applications lie in gray-scale image processing [8,23,30].

Recent studies have also shown that (algebraic) erosions based on infima of implications and their dual dilations also lie at the core of fuzzy morphological associative memories (FMAMs) and that many distributive (matrix) fuzzy associative memory (FAM) models can be viewed as morphological models [31]. This observation also provides a close link to fuzzy relational equations [9,34].

Unlike the aforementioned fuzzy inclusion measure, Kosko's subsethood measure [19] neither commutes with the infimum operator nor satisfies the heritage property. The latter property can be viewed as an advantage rather than a drawback for many applications where the result 0 for crisp $A \not\subseteq B$ is not desirable. Recently, the Kosko subsethood measure was used to devise the Kosko subsethood FAM, a non-distributive two-layer associative memory with competitive hidden nodes [28,29]. The Kosko subsethood measure also belongs to the class of "inclusion" measures that were employed in the fuzzy lattice neurocomputing models of Kaburlasos et al. (note that the "inclusion" measures defined by Kaburlasos et al. in an arbitrary complete lattice differ from the usual fuzzy inclusion measures that satisfy the heritage property) [13,15].

Moreover, the Kosko subsethood complies with Young's well-known axiomatic characterization of a fuzzy subsethood measure but it does not fit into the framework of Sinha and Dougherty [26,37]. In this paper, we follow a less restrictive approach towards subsethood measures that was developed by Fan et al. [10]. Based on this approach, we introduce generalizations of the KS-FAM model called subsethood FAMs (S-FAMs). Given arbitrary subsethood measures, we furthermore derive similarity measures that give rise to new two-layer FAM models named similarity measure FAMs (SM-FAMs). After characterizing the properties of S-FAMs and SM-FAMs, we provide some experimental results including an application to the problem of speaker identification from voice signals without any constraints on what the individual is saying [5,11,32]. In comparison to some competitive classifiers such as k-nearest neighbors, the multi-layer perceptron, and the support vector machine, our new FAM models required a low computational effort and exhibited very satisfactory identification rates.

2 Fuzzy Subsethood and Similarity Measures

The concept of *subsethood* should measure the degree to which a certain fuzzy set A is contained in another fuzzy set B. Let $\mathcal{F}(\mathbf{X})$ denote the class of fuzzy sets over a universe $\mathbf{X}$. The first notion of fuzzy subsethood or inclusion is due to Zadeh [38] who proposed a binary relation $Inc_Z : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \to \{0,1\}$ between fuzzy sets by defining

$$Inc_Z(A,B) = \begin{cases} 1\,, & \text{if } A \subseteq B \\ 0\,, & \text{if } A \not\subseteq B \end{cases}\,, \quad \forall A, B \in \mathcal{F}(\mathbf{X})\,. \tag{1}$$

Unfortunately, this definition "does not do justice to spirit of fuzzy set theory" [6] and represents "an unconscious step backwards in the realm of dichotomy" [1]. Note that the restriction of Inc_Z to $\mathcal{P}(\mathbf{X}) \times \mathcal{P}(\mathbf{X})$ yields the crisp measure of inclusion Inc (i.e. $Inc(A,B) = 1$ if $A \subseteq B$ and 0 if $A \not\subseteq B$ for all $A, B \in \mathcal{P}(\mathbf{X})$) [38]. In general, we refer to a mapping $Inc_{\mathcal{F}} : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \to [0,1]$ as a fuzzy inclusion if $Inc_{\mathcal{F}}$ extends Inc to $\mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X})$. Formally, we have

Definition 1 (Fuzzy Inclusion). *A mapping $Inc_{\mathcal{F}} : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \to [0,1]$ is said to be a* fuzzy inclusion *if $Inc_{\mathcal{F}}(A,B) = Inc(A,B)$ for all $A, B \in \mathcal{P}(\mathbf{X})$.*

In view of the fact that the unit interval forms a complete lattice [3], an infinite number of fuzzy inclusions can be derived from fuzzy implication operators by means of Equation 2. In this context, recall that a complete lattice is given by a set $\mathbb{L}$ such that every subset has an infimum and a supremum in $\mathbb{L}$. The infimum and supremum operators are respectively denoted using the $\bigwedge$ and $\bigvee$ symbols.

Lemma 1. *[1] For every fuzzy implication I, a fuzzy inclusion $Inc_I : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \to [0,1]$ arises via the following definition:*

$$Inc_I(A,B) = \bigwedge_{\mathbf{x} \in \mathbf{X}} I(\mu_A(\mathbf{x}), \mu_B(\mathbf{x})) \quad \forall A, B \in \mathcal{F}(\mathbf{X})\,. \tag{2}$$

Although fuzzy inclusion measures play an important role in fuzzy mathematical morphology where they are used to define fuzzy erosions [8,23,30], one might argue that $Inc_{\mathcal{F}}(A,B)$ is too rigid to adequately express the degree to which A is contained in B [4]. As an illustrative example, consider the following subsets of $\mathbb{Z}$:

$$A = \{x \in \mathbb{Z} \mid x > 0\}\,,\ B = \{x \in \mathbb{Z} \mid x \geq 0\}\,, \text{and } C = \{x \in \mathbb{Z} \mid x < 0\}\,. \tag{3}$$

Note that $Inc_{\mathcal{F}}(B,A) = Inc_{\mathcal{F}}(C,A) = 0$ for every fuzzy inclusion measure $Inc_{\mathcal{F}}$, although it seems reasonable to claim that, in comparison to C, the crisp set B exhibits a larger degree of subsethood in A. Observations like these have led several researchers to put forth sets of axioms describing the requirements that a fuzzy subsethood measure should satisfy. In this paper, we adopt the axiomatic characterization of a subsethood measure that was proposed by Fan et al. [10].

Definition 2 (Subsethood). *A function $S : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \to [0,1]$ satisfying the following properties for all $A, B, C \in \mathcal{F}(\mathbf{X})$ is said to be a* subsethood *measure:*

1. *If $A \subseteq B$ then $S(A,B) = 1$;*
2. *$S(X, \emptyset) = 0$;*
3. *If $A \subseteq B \subseteq C$ then $S(C,A) \leq S(B,A)$ and $S(C,A) \leq S(C,B)$.*

The following proposition provides formulas for constructing subsethood measures.

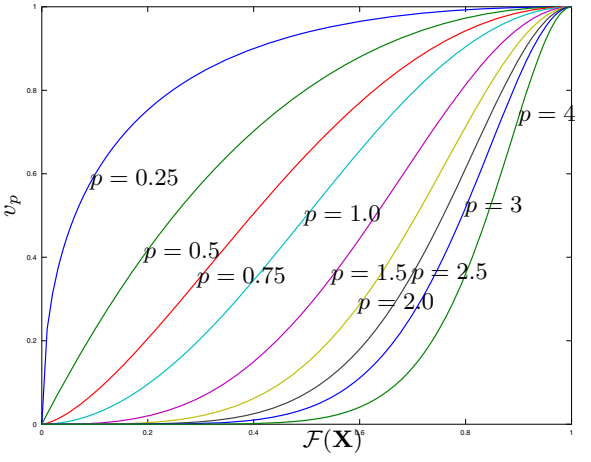


Fig. 1. The function υ_p for $p = 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5, 3, 4$

Proposition 1. *Consider the operators of intersection and union of fuzzy sets. If* $\upsilon : \mathcal{F}(\mathbf{X}) \to [0, \infty)$ *is an arbitrary increasing function such that*

$$\upsilon(A) = 0 \Leftrightarrow A = \emptyset \quad \forall\, A \in \mathcal{F}(\mathbf{X}), \tag{4}$$

then the following operators $\bar{S}$ *and* $\hat{S}$ *represent subsethood measures:*

(a)
$$\bar{S}(A,B) = \begin{cases} 1, & \text{if } A = \emptyset, \\ \frac{\upsilon(A \cap B)}{\upsilon(A)}, & \text{if } A \neq \emptyset, \end{cases} \tag{5}$$

(b)
$$\hat{S}(A,B) = \begin{cases} 1, & \text{if } A = B = \emptyset, \\ \frac{\upsilon(B)}{\upsilon(A \cup B)}, & \text{otherwise.} \end{cases} \tag{6}$$

For a finite universe $\mathbf{X}$, an entire family of subsethood measures arises from the family of functions $\upsilon_p : \mathcal{F}(\mathbf{X}) \to \mathbb{R}$, where $p > 0$, that we propose below. Note that each function υ_p, where $p > 0$, satisfies the requirements of Proposition 1. Figure 1 illustrates υ_p for $\mathcal{F}(\mathbf{X}) = [0, 1]$ for various values of p.

$$\upsilon_p(A) = \sum_{\mathbf{x} \in \mathbf{X}} \frac{1 - \cos(\pi[\mu_A(\mathbf{x})]^p)}{2} \quad \forall A \in \mathcal{F}(\mathbf{X}). \tag{7}$$

Another, simpler example of an increasing function $\upsilon : \mathcal{F}(\mathbf{X}) \to \mathbb{R}$ that satisfies Equation 4 is given by $M(A)$, the cardinality of a fuzzy set $A \in \mathcal{F}(\mathbf{X})$ [24]. For a finite universe $\mathbf{X}$, $M(A)$ is given by

$$M(A) = \sum_{\mathbf{x} \in X} \mu_A(\mathbf{x}). \tag{8}$$

For $\upsilon = M$, Equations 5 and 6 describe respectively the well-known Kosko subsethood measure, denoted using the symbol S_k [18,19], and Willmott's subsethood measure, denoted using the symbol S_L [35]. For the discrete case, Kosko revealed the relationship between $S_k(A, B)$ and the orthogonal projection of A onto $[\emptyset, B] = \{B' \in \mathcal{F}(\mathbf{X}) \mid B' \subseteq B\}$ in accordance to the geometric idea that the degree to which A is contained in B is inversely proportional to the (possibly normalized) distance of A from its orthogonal projection onto $[\emptyset, B]$. Specifically, the subsethood measure of Kosko can be written as follows:

$$S_k(A, B) = \begin{cases} 1, & \text{if } A = \emptyset\,, \\ 1 - \frac{\|\mu_A - \mu_{B(A)}\|_1}{\|\mu_A\|_1}, & \text{if } A \neq \emptyset\,. \end{cases} \tag{9}$$

Here, $B(A)$ denotes the orthogonal projection of A onto $[\emptyset, B]$ [18].

Suppose that in Proposition 1 we have that υ constitutes a *valuation function* in the complete lattice $\mathcal{F}(\mathbf{X})$, where the operations of infimum and supremum are respectively given by the union and intersection of fuzzy sets. In other words, assume that υ satisfies $\upsilon(A \cup B) = \upsilon(A) + \upsilon(B) - \upsilon(A \cap B)$ for all $A, B \in \mathcal{F}(\mathbf{X})$ [3]. In this case, the subsethood measures $\bar{S}$ e $\hat{S}$ also represent "inclusion" measures in the sense of Kaburlasos et. al. (Prop. 4 of [13]). In particular, Kosko's and Willmot's subsethood measures belong to this category since Equation 8 yields a positive valuation function on a finite universe $\mathbf{X}$. Recall that a valuation function υ is called positive if $x < y$ implies that $\upsilon(x) < \upsilon(y)$. Moreover, the functions υ_p introduced in Equation 7 also represent positive valuation functions.

Apart from "inclusion" measures of this type, Kaburlasos et al. also derive distance measures from positive valuation functions. The fuzzy lattice neurocomputing models of Kaburlasos et al. rely heavily on the use of distance and so called "inclusion" measures that are based on positive valuation functions [13,14,15]. We follow a somewhat related, but different approach in our FAM models. Besides the aforementioned subsethood measures, we employ similarity measures that we generate from subsethood measures by means of Theorem 1.

Let us first review the concept of similarity of two fuzzy sets. Note that this concept can be interpreted as a "fuzzification of equality" [22].

Definition 3 (Similarity Measure). *A function $SM : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \to [0, 1]$ is called a* similarity measure*, if SM satisfies the following properties for all $A, B, C \in \mathcal{F}(\mathbf{X})$:*

1. $SM(A, B) = SM(B, A)$;
2. $SM(A, A^c) = 0\ \forall A \in \mathcal{P}(\mathbf{X})$;
3. $SM(A, A) = 1$;
4. *If $A \subseteq B \subseteq C$ then $SM(A, B) \geq SM(A, C)$ and $SM(B, C) \geq SM(A, C)$.*

In the following theorem, we construct the type of similarity measures that we will use in the SM-FAM models of the next section.

Theorem 1. *Let t be a t-norm and let $S : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \to [0, 1]$ be a subsethood measure such that $S(A, A^c) = 0$ for all $A \in \mathcal{P}(\mathbf{X})$. The following operator $SM_S : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \to [0, 1]$ yields a similarity measure:*

$$SM_S(A, B) = S(A, B)\, t\, S(B, A)\ \ \forall A, B \in \mathcal{F}(\mathbf{X})\,. \tag{10}$$

Note that Theorems 1 and 2 presented by Zeng et. al. in [39] match the special cases of the last theorem for product and minimum t-norms, respectively.

3 Subsethood and Similarity Measure FAMs

The purpose of an associative memory (AM) is to store a set of pattern associations $\{(\mathbf{x}^\xi, \mathbf{y}^\xi) \in \mathbf{X} \times \mathbf{Y} : \xi = 1, \ldots, k\}$, also known as the fundamental memory set, such that the desired output pattern $\mathbf{y}^\gamma$ can be retrieved upon presentation of a possibly noisy or incomplete version of an input pattern $\mathbf{x}^\gamma$. Formally speaking, an AM describes a map $\phi : \mathbf{X} \to \mathbf{Y}$ that associates $\mathbf{x}^\xi$ with $\mathbf{y}^\xi$ for all $\xi = 1, \ldots, k$. Ideally, we have for all $\xi = 1, \ldots, k$ that $\phi(\mathbf{x}^\xi) = \mathbf{y}^\xi$ and, in addition, $\phi(\tilde{\mathbf{x}}^\xi) = \mathbf{y}^\xi$ for corrupted or noisy versions $\tilde{\mathbf{x}}^\xi$ of input patterns $\mathbf{x}^\xi$. In practice, many AM models are only able to store a subset of the fundamental memories and corrupted or noisy versions can only be retrieved approximately, i.e., $\phi(\tilde{\mathbf{x}}^\xi) \simeq \mathbf{y}^\xi$. If the pairs $(\mathbf{x}^\xi, \mathbf{y}^\xi)$ reside in $\mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{Y})$, where $\mathbf{X}$ and $\mathbf{Y}$ are arbitrary universes, we refer to the AM given by $\phi : \mathcal{F}(\mathbf{X}) \to \mathcal{F}(\mathbf{Y})$ as a fuzzy associative memory (FAM).

In this section, we introduce three types of FAM models based on subsethood measures. In the following definitions, we consider a set of associations $\{(A^\xi, B^\xi) \in \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{Y}) \mid \xi = 1, \ldots, k\}$. Moreover, let $S : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \to [0, 1]$ be an arbitrary subsethood measure.

Definition 4 (S-FAM). *For $A \in \mathcal{F}(\mathbf{X})$ consider the index set*

$$I(A) = \{j \in \{1, \ldots, k\} : S(A^j, A) = \max_{\xi=1,\ldots,k} S(A^\xi, A)\}. \tag{11}$$

The subsethood FAM based on S*, for short* S-FAM*, is given by the following mapping* $\mathcal{M} : \mathcal{F}(\mathbf{X}) \to \mathcal{F}(\mathbf{Y})$*:*

$$\mathcal{M}(A) = \bigcup_{j \in I(A)} B^j. \tag{12}$$

Definition 4 generalizes the Kosko subsethood FAM (KS-FAM) that we introduced in [28] and that we successfully applied to the problem of vision-based self-localization in robotics [29]. More precisely, S is given by the Kosko subsethood in the KS-FAM model.

An inversion of the order in which A^j and A as well as A^ξ and A appear in Equation 11 leads to the dual S-FAM model.

Definition 5 (Dual S-FAM). *For $A \in \mathcal{F}(\mathbf{X})$ consider the index set*

$$J(A) = \{j \in \{1, \ldots, k\} : S(A, A^j) = \max_{\xi=1,\ldots,k} S(A, A^\xi)\}. \tag{13}$$

The dual subsethood FAM based on S*, for short* dual S-FAM*, is given by the following mapping* $\mathcal{W} : \mathcal{F}(\mathbf{X}) \to \mathcal{F}(\mathbf{Y})$*:*

$$\mathcal{W}(A) = \bigcup_{j \in J(A)} B^j. \tag{14}$$

Note that the S-FAM and dual S-FAM models are well-defined because every component of $\mathcal{M}(A)$ and $\mathcal{W}(A)$ is given by a maximum over a finite set. The following theorem establishes conditions for the perfect recall of the fundamental memories:

Theorem 2. *Let $\mathcal{M}$ e $\mathcal{W}$ represent respectively the S-FAM and the dual S-FAM based on the subsethood measure S. If the set $\{(A^\xi, B^\xi) \in \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{Y}) \mid \xi = 1, \ldots, k\}$ satisfies $S(A^\xi, A^\gamma) < 1$ for all $\xi \neq \gamma$, then we have*

$$\mathcal{M}(A^\xi) = \mathcal{W}(A^\xi) = B^\xi \ \ \forall \xi = 1, \ldots, k\,. \tag{15}$$

The next result deals with the error correction capabilities of the S-FAM and dual S-FAM models.

Theorem 3. *Let S be a subsethood measure that is continuous in both arguments with respect to an arbitrary given metric d. Let the symbols $\mathcal{M}$ e $\mathcal{W}$ denote respectively the S-FAM and the dual S-FAM based on S and let $\{(A^\xi, B^\xi) \in \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{Y}) \mid \xi = 1, \ldots, k\}$ represent the fundamental memory set. If $S(A^\xi, A^\gamma) < 1$ for all $\xi \neq \gamma$, then there exist $\delta_\mathcal{M}, \delta_\mathcal{W} > 0$ such that the following statements are valid for all $\gamma = 1, \ldots, k$ and for all $A \in \mathcal{F}(\mathbf{X})$:*

(i) $d(A, A^\gamma) < \delta_\mathcal{M}$ *implies that* $\mathcal{M}(A) = B^\gamma$;
(ii) $d(A, A^\gamma) < \delta_\mathcal{W}$ *implies that* $\mathcal{W}(A) = B^\gamma$.

It can be shown that $\delta_\mathcal{M}$ and $\delta_\mathcal{W}$ in Theorem 3 depend on $\beta = \max_{\xi \neq \gamma} S(A^\xi, A^\gamma)$ and that smaller values of β lead to larger values of $\delta_\mathcal{M}$ and $\delta_\mathcal{W}$. Furthermore, lower values of β can be achieved by normalizing the patterns A^ξ. Recall that a normalization strategy was successfully applied in the context of Kosko subsethood FAMs for a discrete domain $\mathbf{X}$ [29]. More precisely, the following function $\Phi : [0,1]^n \to [0,1]^n$ was defined in terms of its coordinates $\Phi(\mathbf{x})_i$ for all $i = 1, \ldots, n$:

$$\Phi(\mathbf{x})_i = \begin{cases} x_i & \text{, if } x_i = 0 \text{ or } x_i = 1 \\ \dfrac{x_i - m_\mathbf{x} + 1.5}{3} & \text{, otherwise.} \end{cases} \tag{16}$$

Here, the symbol $m_\mathbf{x}$ denotes the mean of $\mathbf{x} \in [0,1]^n$ over the restricted domain $I = \{i : 0 < x_i < 1\}$, that is

$$m_\mathbf{x} = \frac{\sum_{i \in I} x_i}{\mid I \mid}.$$

Let us finish this section by introducing an FAM based on a similarity measure.

Definition 6 (SM-FAM). *Let $SM : \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{X}) \to [0,1]$ be a similarity measure. Let us define the following index set for $A \in \mathcal{F}(\mathbf{X})$:*

$$K(A) = \{j \in \{1, \ldots, k\} : SM(A, A^j) = \max_{\xi = 1, \ldots, k} SM(A, A^\xi)\}. \tag{17}$$

The similarity measure FAM based on SM*, for short SM-*FAM*, is given by the following mapping $\mathcal{Q} : \mathcal{F}(\mathbf{X}) \to \mathcal{F}(\mathbf{Y})$:*

$$\mathcal{Q}(A) = \bigcup_{j \in K(A)} B^j. \tag{18}$$

Perfect recall conditions for the SM-FAM can be established in a similar way as for the S-FAM and dual S-FAM models:

Theorem 4. *Let $\mathcal{Q}$ represent an SM-FAM based on a similarity measure SM. If the set $\{(A^\xi, B^\xi) \in \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{Y}) \mid \xi = 1, \ldots, k\}$ satisfies $SM(A^\xi, A^\gamma) < 1$ for all $\xi \neq \gamma$, then we have*

$$\mathcal{Q}(A^\xi) = B^\xi \ \forall \xi = 1, \ldots, k\,. \tag{19}$$

If the similarity measure SM is induced by a subsethood measure S in the sense of Equation 10 then we obtain the following theorem about SM-FAMs that is similar to Theorem 3.

Theorem 5. *Consider a subsethood measure S that is continuous in both arguments with respect to an arbitrary given metric d. Let SM_S be the similarity measure given by $SM_S(A, B) = S(A, B)\ t\ S(B, A)$ ($\forall\ A, B \in \mathcal{F}(\mathbf{X})$) where t stands for an arbitrary continuous t-norm. Moreover, let $\mathcal{Q}$ denote the SM-FAM based on SM_S and let $\{(A^\xi, B^\xi) \in \mathcal{F}(\mathbf{X}) \times \mathcal{F}(\mathbf{Y}) \mid \xi = 1, \ldots, k\}$ represent the fundamental memory set. If $S(A^\xi, A^\gamma) < 1$ for all $\xi \neq \gamma$, then there exists a value $\delta_\mathcal{Q} > 0$ such that, for all $\gamma = 1, \ldots, k$ and for all $A \in \mathcal{F}(\mathbf{X})$, $d(A, A^\gamma) < \delta_\mathcal{Q}$ implies that $\mathcal{Q}(A) = B^\gamma$.*

In the following section, we will use the S-FAM, dual S-FAM, and SM-FAM models in simulations concerning independent speaker identification from voice signals.

4 Application to the Problem of Speaker Identification

In order to evaluate the performance of the novel FAM models in a real-world problem, let us consider a text-independent, closed-set automatic speaker identification (ASI) application. Briefly, the task of ASI is to identify a speaker on the basis of a set of utterances. We speak of a closed-set problem if the unknown individual belongs to a pre-existing pool of speakers and therefore new individuals are not subjected to identification. In a text-independent problem, the recognition procedure should work regardless of the underlying spoken words by taking into account general characteristics of the person's voice. Applications of ASI include access control of information services, banking transactions via telephone networks, and automatic speaker detection in speech dialog or recorded meetings [5,11,32].

In our experiments, we used the Brazilian Portuguese speech database elaborated by Ynoguti and Violaro [36] that corresponds to 9 male and 9 female speakers. For each speaker, the database includes a set of 10 different words, chosen to meet applications such as computer/machine operations and banking services. Every word in the entire database is said at most once by each person but the same word can be uttered by different speakers. For example, the Portuguese acronym "CDB", which refers to a certificate of deposit, appears 3 times in the database and the phonetically very similar acronym "CDC", a Brazilian banking expression that refers to a consumer's preapproved amount of credit, appears twice in the database. For each speaker, we used 5 word samples for training and 5 distinct samples for testing.

The Mel Frequency Cepstral Coefficients (MFCCs), as well as their velocity and acceleration, have been used as acoustic features for the identification of the speakers.

Table 1. Classification errors in the ASI application produced by S-FAM, dual S-FAM, and SM-FAM models based on subsethood measures given by Equation 5 and υ_p given by Equation 7

p	0.50	1.00	1.50	2.00	2.50	3.00
S-FAM	**23.33**	24.44	30.00	32.22	30.00	34.44
Dual S-FAM	28.89	**25.56**	27.78	28.89	30.00	32.22
SM-FAM	25.56	25.56	26.67	23.33	**22.22**	23.33

Specifically, each digitized waveform signal of an utterance has been submitted to the seven steps described by Jurafisky and Martin [12]. This procedure yields a variable sequence of feature vectors $\mathbf{f}^\xi$ of length 36, where the index ξ depends on the speaker, the spoken word, and the time window. This set of feature vectors was divided into a training set and a test set consisting of 4816 and 5101 samples, respectively. Both training and test sets, as well as some MATLAB® source codes used in this application, are available at [33].

Recall that MFCCs have been widely used in speaker recognition tasks due to the following reasons: The coefficients tend to be uncorrelated and they retain information solely about the vocal tract filter of the speaker. In contrast to Jurafisky and Martin, our feature vectors do not take into account any energy information because this information is mainly correlated with the type of phoneme. We did however employ information concerning the energy to discard silent or noisy frames from the utterance signal.

During the training phase, each feature vector $\mathbf{f}^\xi$ was converted into a discrete fuzzy set $A^\xi \in [0,1]^{36}$. If $\hat{f}$ and $\check{f}$ denote respectively $\bigvee_{\xi=1} \bigvee_{i=1}^{36} f_i^\xi$ and $\bigwedge_{\xi=1} \bigwedge_{i=1}^{36} f_i^\xi$ then we computed $A^\xi \in [0,1]^{36}$ as follows using the function Φ given by Equation 16:

$$A^\xi = \Phi\left(\left[\frac{\mathbf{f}^\xi - \check{f}}{\hat{f} - \check{f}} \wedge 1\right] \vee 0\right). \tag{20}$$

Also, each fuzzy set A^ξ was associated to a vector (crisp set) $B^\xi \in \{0,1\}^{18}$ which is zero except at the component that corresponds to the speaker label. Hence, we obtained a fundamental memory set $\{(A^\xi, B^\xi) : \xi = 1, \ldots, 4816\}$ that was stored in the FAMs.

In the testing phase, we presented a waveform signal produced by an unknown speaker to the system. First, we extracted a sequence $\{\mathbf{f}^\gamma\}$ of feature vectors from the signal. For each γ, the vector $\mathbf{f}^\gamma$ was converted into a discrete fuzzy set A^γ that we fed as input into the FAM models. Then, the resulting outputs B^γ's were converted into labels ℓ^γ representing the speakers. Finally, the utterance was attributed to the speaker whose label occurred most frequently in the sequence $\{\ell^\gamma\}$.

We employed Equations 5 and 6 with functions υ_p defined in Equation 7 to generate subsethood and similarity measures. The latter were constructed using the minimum t-norm in Equation 10. We varied $p = 0.5, 1, \ldots, 3$ and applied the resulting S-FAM, dual S-FAM, and SM-FAM models to the ASI problem. Tables 1 and 2 display the resulting classification errors.

For comparative purposes, we present the classification errors produced by some very well-known classifiers in Table 3. The best overall recognition rate of 83.33% in

Table 2. Classification errors in the ASI application produced by S-FAM, dual S-FAM, and SM-FAM models based on subsethood measures given by Equation 6 and υ_p given by Equation 7

p	0.50	1.00	1.50	2.00	2.50	3.00
S-FAM	**23.33**	24.44	28.89	33.33	30.00	36.67
Dual S-FAM	28.89	**25.56**	27.78	30.00	36.67	56.67
SM-FAM	25.56	25.56	26.67	**22.22**	23.33	23.33

Table 3. Classification errors produced by some classifiers in ASI experiment

Classifier	SVM	MLP_{500}	MLP_{360}	1-NN	3-NN	5-NN	7-NN	9-NN
Error (%)	16.67	24.45	32.45	26.67	28.89	30.00	30.00	31.11

the testing phase was achieved by a support vector machine (SVM) with a Gaussian radial basis kernel with spread $\sigma^2 = 1$. A multi-layer perceptron (MLP) with 500 hidden neurons equipped with hyperbolic tangent activation functions and linear output neurons yielded an average recognition rate of $75.55\ \%$ for the test data. A similar MLP with 360 hidden neurons correctly classified an average of 67.55% of the test data. In both MLP models, we separated 30% of the training data for use as validation data in conjunction with early stopping so as to avoid overfitting. Training was performed using one-step secant backpropagation [2] as well as MATLAB®'s default pre-processing and initialization routines.

Finally, we tested the k-nearest neighbor (k-NN) models for $k = 1, 3, \ldots, 9$ with the usual Euclidean distance measure and obtained the smallest classification error of 26.67% for $k = 1$. Table 3 summarizes these results. Here, the MLPs with 500 and 360 hidden neurons are respectively referred to as MLP_{500} and MLP_{360}.

Note that the best error rates of 23.33, 25.56, and 22.22 percent that were produced by the S-FAM, dual S-FAM, and SM-FAM models, respectively, using either one of Equations 5 and 6, compare favorably to the classification results of the MLP and k-NN models. The SVM model, that outperformed all other models in this experiment, is computationally far more expensive than the S-FAM, dual S-FAM, and SM-FAM models introduced in this paper. Here, an application of the SVM involves solving 18 quadratic problems of size 4816 by means of an active constraints method during the training phase [25].

5 Concluding Remarks

In this paper, we introduced the subsethood FAMs (S-FAMs), that generalize the recently defined KS-FAM, as well as dual subsethood FAMs (dual S-FAMs). Both of these models are two-layer non-distributive models in which a certain degree of subsethood of one fuzzy set in another one is evaluated at each hidden node. Given a certain subsethood measure, we showed how to derive a similarity measure that we used for

constructing so called similarity measure FAM (SM-FAM). We also provided a method for generating an infinite family of parameterized subsethood measures that give rise to an infinite family of S-FAMs, dual S-FAMs, and SM-FAMs. Concerning these three types of associative memories, we provided sufficient conditions for perfect recall as well as theoretical results on the minimal amount of tolerance with respect to arbitrary noise.

Finally, we successfully applied the S-FAM, dual S-FAM, and the SM-FAM models to a problem of text-independent, closed-set automatic speaker identification (ASI). All three models, in particular the SM-FAM, exhibited competitive performances in comparison to other models.

References

1. Bandler, W., Kohout, L.: Fuzzy power sets and fuzzy implication operators. Fuzzy Sets and Systems 4(1), 13–30 (1980)
2. Battiti, R.: First and second-order methods for learning: between steepest descent and Newton's method. Neural Computation 4, 141–166 (1992)
3. Birkhoff, G.: Lattice Theory, 3rd edn. American Mathematical Society, Providence (1993)
4. Bloch, I., Maitre, H.: Fuzzy mathematical morphologies: a comparative study. Pattern Recognition 28(9), 1341–1387 (1995)
5. Campbell, J.: Speaker recognition: A tutorial. Proceedings of the IEEE 85(9), 1437–1462 (1997)
6. Cornelis, C., der Donck, C.V., Kerre, E.: Sinha-Dougherty approach to the fuzzification of set inclusion revisited. Fuzzy Sets and Systems 134(2), 283–295 (2003)
7. De Baets, B., Kerre, E., Gupta, M.: The fundamentals of fuzzy mathematical morphology, part 1: basic concepts. International Journal of General Systems 23, 155–171 (1994)
8. Deng, T.Q., Heijmans, H.J.A.M.: Grey-scale morphology based on fuzzy logic. Journal of Mathematical Imaging and Vision 16(2), 155–171 (2002)
9. Di Nola, A., Sessa, S., Pedrycz, W., Sanchez, E.: Fuzzy Relation Equations and Their Applications to Knowledge Engineering. Kluwer Academic Publishers, Norwell (1989)
10. Fan, J., Xie, W., Pei, J.: Subsethood measure: new definitions. Fuzzy Sets and Systems 106(2), 201–209 (1999)
11. Gish, H., Schmidt, M.: Text-independent speaker identification. IEEE Signal Processing Magazine 11(4), 18–32 (1994)
12. Jurafsky, D., Martin, J.: Speach and Language Processing: An Introduction to Natural Language Processing. In: Computational Linguistics, and Speech Recognition, 2nd edn. Prentice Hall Series in Artificial Intelligence, Prentice Hall, Upper Saddle River (2009)
13. Kaburlasos, V.G., Athanasiadis, I.N., Mitkas, P.A.: Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation. International Journal of Approximate Reasoning 45(1), 152–188 (2007)
14. Kaburlasos, V.G., Papadakis, S.E.: A granular extension of the fuzzy-artmap (FAM) neural classifier based on fuzzy lattice reasoning (FLR). Neurocomputing 72(10-12), 2067–2078 (2009)
15. Kaburlasos, V.G., Petridis, V.: Fuzzy lattice neurocomputing (FLN) models. Neural Networks 13(10), 1145–1170 (2000)
16. Kaburlasos, V.G., Moussiades, L., Vakali, A.: Fuzzy lattice reasoning (FLR) type neural computation for weighted graph partitioning. Neurocomput. 72, 2121–2133 (2009)
17. Kitainik, L.: Fuzzy Decision Procedures with Binary Relations. Kluwer Academic Publishers (1993)

18. Kosko, B.: Fuzziness vs. probability. Int. J. General Systems 17, 211–240 (1990)
19. Kosko, B.: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice Hall, Englewood Cliffs (1992)
20. Kuncheva, L.I.: Fuzzy rough sets: Application to feature selection. Fuzzy Sets and Systems 51(2), 147–153 (1992)
21. Lee, H.M., Wang, W.T.: A neural network architecture for classification of fuzzy inputs. Fuzzy Sets Syst. 63, 159–173 (1994)
22. Liu, X.: Entropy, distance measure and similarity measure of fuzzy sets and their relations. Fuzzy Sets Syst. 52, 305–318 (1992)
23. Nachtegael, M., Kerre, E.E.: Connections between binary, gray-scale and fuzzy mathematical morphologies. Fuzzy Sets and Systems 124(1), 73–85 (2001)
24. Pedrycz, W., Gomide, F.: Fuzzy Systems Engineering: Towards Human-Centric Computing. Wiley, IEEE Press, New York (2007)
25. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2001)
26. Sinha, D., Dougherty, E.R.: Fuzzification of set inclusion: theory and applications. Fuzzy Sets and Systems 55(1), 15–42 (1993)
27. Sinha, D., Sinha, P., Dougherty, E.R., Batman, S.: Design and analysis of fuzzy morphological algorithms for image processing. IEEE Transactions on Fuzzy Systems 5(4), 570–583 (1997)
28. Sussner, P., Esmi, E.: An Introduction to the Kosko Subsethood FAM. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 343–350. Springer, Heidelberg (2010)
29. Sussner, P., Esmi, E.L., Villaverde, I., Graña, M.: The Kosko subsethood fuzzy associative memory (KS-FAM): Mathematical background and applications in computer vision. Journal of Mathematical Imaging and Vision (2011)
30. Sussner, P., Valle, M.E.: Classification of fuzzy mathematical morphologies based on concepts of inclusion measure and duality. Journal of Mathematical Imaging and Vision 32(2), 139–159 (2008)
31. Sussner, P., Valle, M.E.: Fuzzy associative memories and their relationship to mathematical morphology. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, ch. 33, John Wiley and Sons, Inc., New York (2008)
32. Togneri, R., Pullella, D.: An overview of speaker identification: Accuracy and robustness issues. IEEE Circuits and Systems Magazine 11(2), 23–61 (2011)
33. Valle, M.E., Sakuray, F.: Database and MATLAB® source codes for ASI using MFCCs. Center for Exact Sciences, University of Londrina, Brazil (2011), http://www.uel.br/pessoal/valle/Codes/Speaker_Recognition.zip
34. Valle, M.E., Sussner, P.: A general framework for fuzzy morphological associative memories. Fuzzy Sets and Systems 159(7), 747–768 (2008)
35. Willmott, R.: On the transitivity of containment and equivalence in fuzzy power set theory. Journal of Mathematical Analysis and Applications 120(1), 384–396 (1986)
36. Ynoguti, C., Violaro, F.: A Brazilian Portuguese speech database. In: Anais do XXVI Simpósio Brasileiro de Telecomunicações (SBrT 2008). Rio de Janeiro, Brasil (September 2008)
37. Young, V.R.: Fuzzy subsethood. Fuzzy Sets and Systems 77(3), 371–384 (1996)
38. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
39. Zeng, W., Li, H.: Inclusion measures, similarity measures, and the fuzziness of fuzzy sets and their relations. International Journal of Intelligent Systems 21(6), 639–653 (2006)

A Novel Lattice Associative Memory Based on Dendritic Computing

Gerhard X. Ritter[1], Darya Chyzhyk[2,⋆], Gonzalo Urcid[3,⋆⋆], and Manuel Graña[2]

[1] CISE Department, University of Florida, Gainesville, FL 32611–6120, USA
ritter@cise.ufl.edu
[2] Computational Intelligence Group, University of the Basque Country, Spain
{darya.chyzhyk,manuel.grana}@ehu.es
[3] Optics Department, INAOE, Tonantzintla, Pue. 72000, Mexico
gurcid@inaoep.mx

Abstract. We present a novel hetero-associative memory based on dendritic neural computation. The computations in this model are based on lattice group operations. The proposed model does not suffer from the usual storage capacity problem and is extremely robust in the presence of various types of noise and data corruption.

Keywords: Dendritic Computing, Hetero-associative memory, Lattice algebra, Data Corruption.

1 Introduction

The concept of an associative memory is a fairly intuitive one as it is based on the observation that an associative memory seems to be one of the primary functions of the brain. We easily *associate* the face of a friend with that of the friend's name, or a name with a telephone number. For this reason artificial neural networks (ANNs) that are capabable of storing several types of patterns and corresponding associations are referred to as *associative memories*. Such memories retrieve stored associations when presented with corresponding input patterns. An associative memory is said to be *robust in the presence of noise* if presented with a corrupted version of a prototype input pattern it is still capable of retrieving the correct association.

In classical pattern recognition, patterns are viewed as column vectors in Euclidean space. Each component of a pattern vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)' \in \mathbb{R}^n$ corresponds to one of the pattern's features. The numerical value x_i of a pattern feature can represent a variety of objects or physical features such as signal strength, curvature, a probability value, mean mass, and so on. One goal in the theory of associative memories is for the memory to recall a stored pattern $\mathbf{y} \in \mathbb{R}^m$ when presented a pattern $\mathbf{x} \in \mathbb{R}^n$, where the pattern association expresses some desired pattern correlation. More precisely, suppose $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^K\} \subset$

⋆ Corresponding author.

⋆⋆ G. Urcid is grateful with CONACYT for partial financial support grant # 22036.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 491–502, 2012.

$\mathbb{R}^n$ and $Y = \{\mathbf{y}^1, \ldots, \mathbf{y}^K\} \subset \mathbb{R}^m$ are two sets of pattern vectors with desired association given by the diagonal $\{(\mathbf{x}^\xi, \mathbf{y}^\xi) : \xi = 1, \ldots, K\}$ of $X \times Y$. The goal is to store these pattern pairs in some memory M such that for $\xi = 1, \ldots, K$, M recalls $\mathbf{y}^\xi$ when presented with the pattern $\mathbf{x}^\xi$. If such a memory M exists, then we shall express this association symbolically by $\mathbf{x}^\xi \rightarrow M \rightarrow \mathbf{y}^\xi$. Whenever $X = Y$, then the memory M is called an *auto-associative memory*, otherwise it is called a *hetero-associative* memory or simply an *associative* memory.

The matrix correlation memories resulting from the work of Steinbuch, Kohonen, Anderson, and Hopfield were the earliest artificial neural network (ANN) examples of associative memories [1,2,3,4,5,6,7,8,9,10]. Matrix correlation memories based on lattice computations were first introduced in the late 1990s [11,12,13]. These memories had the advantage of unlimited storage capacity and one step convergence. However, they were susceptible to certain types of random noise. The concept of dendritic computing was partially due to trying to eliminate the noise problem encountered in the construction of artificial memories. The other reason was to provide an artificial neural paradigm that is closer related to actual biological neural computation [14].

Lattice based Neural Networks (LNNs) - although not yet recognized as mainstream in machine learning - have become an integral part of artificial neural network theory [15,16]. One reason for this is their simplicity and fast learning methods and another is due to their successful applicability in several disciplines [17,18,19,20,21,22]. In this paper the focus is on a novel Dendritic Lattice based (hetero) Associative Memory or, simply, DLAM. Recently two new DLAMs have appeared in the literature [23] and [24]. The former being a generalization of the DLAMs given in [25], while the latter had no predecessor within lattice theory. However, the latter model was presented as an auto-associative memory. Here we show that the model easily generalizes to a hetero-associative memory. Similar to earlier lattice based associative memories, this new DLAM has unlimited storage capacity in that it can memorize any finite number of association and provides perfect recall for non-noisy input. However, as we shall demonstrate, its greatest advantage over prior associative memories is that it can recall association even when the input is an exemplar pattern that has been corrupted by more than 90% of random noise.

The remainder of this paper is partitioned into three sections. In Section 2 we provide a brief overview of Dendritic Lattice based Neural Networks (DLNNs). Rationale for the DLNN approach is not discussed as it can be found in [14]. Section 3 introduces the new DLAM model and explains the computations occuring in the different layers as well as the funtion of each layer. The robustness of the DLAM in the presence of various types of noise is demonstrated in Section 4. Conclusions and some pertinent observations are presented in the final section.

2 The Dendritic Lattice Based Model of ANNs

Roughly speaking, a lattice based neural network is an ANN in which the basic neural computations are based on the operations of a lattice ordered group.

By a lattice ordered group we mean a set L with an associated algebraic structure $(L, \vee, \wedge, +)$, where $(L, \vee, \wedge)$ is a lattice and $(L, +)$ is a group with the property that every group translation is isotone; that is, if $x \leq y$, then $a+x+b \leq a+y+b$ $\forall\, a, b \in L$. Given the set $\mathfrak{O} = \{\vee, \wedge, +\}$ of lattice goup operations, then the symbols $\oplus$ and $\otimes$ will mean that $\oplus, \otimes \in \mathfrak{O}$ but are not explicitly specified lattice operations. Similarly, symbols of form $\bigoplus$ and $\bigotimes$ will denote lattice operations derived from the operations $\oplus$ and $\otimes$, respectively. For example, $\bigoplus_{i=1}^{n} a_i = a_1 \oplus a_2 \oplus \cdots \oplus a_n$. Hence, specifying $\oplus = \vee$, then $\bigoplus_{i=1}^{n} a_i = \bigvee_{i=1}^{n} a_i = a_1 \vee a_2 \vee \cdots \vee a_n$.

In the dendritic model of ANNs, a finite set of presynaptic neurons $N_1, \ldots, N_n$ provides information through its axonal arborization to the dendritic trees of some other finite set of postsynaptic neurons $M_1, \ldots, M_m$. The dendritic tree of a postsynaptic neuron M_j is assumed to consist of a finite number of branches $d_{j1}, \ldots, d_{jK_j}$ which contain the synaptic sites upon which the axonal fibers of the presynaptic neurons terminate. The *strength* of the synapse on the kth dendritic branch d_{jk} $(k \in \{1, \ldots, K(j)\})$ which serves as a synaptic site for a terminal axonal branch fiber of N_i is denoted by w_{ijk}^{ℓ} and is also called its *synaptic weight*. The superscript ℓ is associated with the postsynaptic response that is generated within and in close proximity of the synapse. Specifically, $\ell = 0$ and $\ell = 1$ denote an inhibitory or excitatory postsynaptic response, respectively. It is possible for several axonal fibers to synapse on the same or different synaptic sites on a given branch d_{jk}, with the former case implying that $w_{ijk}^{\ell} = w_{hjk}^{\ell}$. The total response (or output) of d_{jk} to the received input at its synaptic sites is given by

$$\tau_k^j(\mathbf{x}) = p_{jk} \bigoplus_{i \in I(k)} \bigotimes_{\ell \in \mathcal{L}(i)} [(-1)^{1-\ell} \left(x_i + w_{ijk}^{\ell}\right)], \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in L^n$ with L^n denoting the n-fold cartesian product of L, $x_i \in L$ denotes the information propagated by N_i via its axon and axonal branches, $\mathcal{L}(i) \subseteq \{0, 1\}$ corresponds to the postsynaptic response generated at the synaptic region to the input received from N_i, and $I(k) \subseteq \{1, \ldots, n\}$ corresponds to the set of all presynaptic neurons with terminal axonal fibers that synapse on the kth dendritic branch of M_j. The value $p_{jk} \in \{-1, 1\}$ marks the final signal outflow from the kth branch as inhibitory if $p_{jk} = -1$ and excitatory if $p_{jk} = 1$. The value $\tau_k^j(\mathbf{x})$ is passed to the cell body of M_j and the state of M_j is a function of the combined values received from its dendritic structure and is given by

$$\tau^j(\mathbf{x}) = p_j \bigotimes_{k=1}^{K_j} \tau_k^j(\mathbf{x}), \tag{2}$$

where K_j denotes the total number of dendritic branches of M_j and $p_j = \pm 1$ denotes the response of the cell to the received input. Here again $p_j = -1$ means rejection (inhibition) and $p_j = 1$ means acceptance (exitation) of the received input. Figure 1(a) illustrates the neural pathways from the presynaptic neurons to the postsynaptic neuron M_j. Figure 1(b) illustrates a dendritic network. The prime example of a lattice ordered group is the set $\mathbb{R}$ of real numbers together

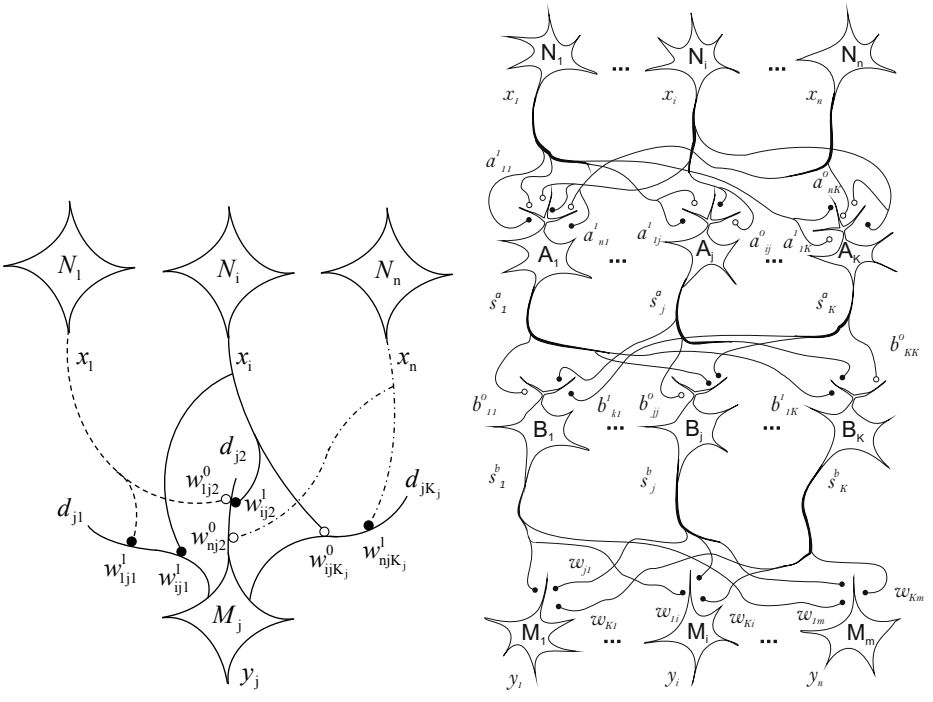


Fig. 1. (a) Terminal branches of axonal fibers originating from the presynaptic neurons make contact with synaptic sites on dendritic branches of M_j. (b) Structure of a dendritic network.

with the binary operations of the maximum ($\vee$) and minimum ($\wedge$) of two numbers and the group operation of addition, denoted by ($\mathbb{R}$, $\vee$, $\wedge$, $+$). It is also the lattice employed in this paper. Thus, for example, eqn.1 could assume the form

$$\tau_k^j(\mathbf{x}) = p_{jk} \bigvee_{i \in I(k)} \bigwedge_{\ell \in \mathcal{L}(i)} (-1)^{1-\ell} \left(x_i + w_{ijk}^{\ell}\right), \tag{3}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, and $x_i \in \mathbb{R}$, while eqn.2 could be of form

$$\tau^j(\mathbf{x}) = p_j \sum_{k=1}^{K_j} \tau_k^j(\mathbf{x}). \tag{4}$$

3 Dendritic Lattice Associative Memories

The Dendritic Lattice based Associative Memory or DLAM described in this section can store any desirable number of pattern associations and has perfect

recall when presented with an exemplary pattern. Furthermore, it is extremely robust in the presence of noise and can be applied to both Boolean and real number value patterns.

The proposed DLAM consists of four layers of neurons: an input layer, two hidden layers, and an output layer. The number of neurons in each layer is predetermined by the dimensionality of the pattern domains. Explicitely, if $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^K\} \subset \mathbb{R}^n$ and $Y = \{\mathbf{y}^1, \ldots, \mathbf{y}^K\} \subset \mathbb{R}^m$, then the number of neurons in the input layer is n, in the two hidden layers it is K, and the number in the output layer is m. We denote the neurons in the input layer by $N_1, \ldots, N_n$, in the first hidden layer by $A_1, \ldots, A_K$, in the second hidden layer by $B_1, \ldots, B_K$ and in the output layer by $M_1, \ldots, M_m$. We refer to the first and second hidden layer as the *A-layer* and the *B-layer*, respectively. For a given input pattern $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, the ith neuron N_i will assume as its value the ith coordinate x_i of $\mathbf{x}$ and will propagate this value through its axonal arborization to the dentrites of the hidden layer neurons. The dendritic tree of each hidden neuron A_j has n single branches $d_{j1}, \ldots, d_{jn}$, and each neuron N_i has two axonal fibers terminating on the synaptic sites located on the corresponding branch d_{ji} of the hidden layer neuron A_j as depicted in Figure 1(b). Observe that in this formulation the dendritic branch counter $k = i$, making the extra counter k unecessary. The two synaptic weights associated with the two synaptic sites of d_{ji} will be denoted by a_{ij}^{ℓ} and defined by $a_{ij}^{\ell} = -x_i^j$ for $\ell = 0, 1$. The output of each dendritic branch is denoted by $\tau_i^j(\mathbf{x})$. Here we use the formula given by eqn. 3 in order to compute this value. Setting $p_{jk} = -1$ and using the fact that $I(k) = I(i) = \{i\}$, eqn. 3 reduces to

$$\begin{aligned}\tau_i^j(\mathbf{x}) &= -\bigwedge_{\ell=0}^{1}(-1)^{1-\ell}\left(x_i + a_{ij}^{\ell}\right) = -[-(x_i - x_i^j) \wedge (x_i - x_i^j)] \\ &= -[-(x_i - x_i^j) \wedge -(x_i^j - x_i)] = (x_i - x_i^j) \vee (x_i^j - x_i). \end{aligned} \tag{5}$$

It follows from eqn. 5 that $\tau_i^j(\mathbf{x}) = 0 \Leftrightarrow x_i = x_i^j$ and $\tau_i^j(\mathbf{x}) > 0 \Leftrightarrow x_i \neq x_i^j$. The value $\tau_i^j(\mathbf{x})$ is passed to the cell body of A_j and its state is a function of the combined values received from its dendritic structure. This state is computed using eqn. 3 with $p_j = 1$. Specifically, we have

$$\tau_A^j(\mathbf{x}) = \sum_{i=1}^{n} \tau_i^j(\mathbf{x}) = \sum_{i=1}^{n}(x_i - x_i^j) \vee (x_i^j - x_i) = \sum_{i=1}^{n} |x_i - x_i^j|. \tag{6}$$

It follows that each neuron A_j in the A-layer computes the L_1-distance between the input pattern $\mathbf{x}$ and the jth exemplar pattern $\mathbf{x}^j$. That is, $\tau_A^j(\mathbf{x}) = d_1(\mathbf{x}, \mathbf{x}^j)$. The activation function for the A-layer neurons is derived from the identity function, namely

$$f_A(z) = \begin{cases} z & \text{if } z \leq T \\ \infty & \text{if } z > T \end{cases}, \tag{7}$$

where T is a user defined threshold. We denote the output of A_j by $s_A^j = f_A(\tau_A^j(\mathbf{x}))$ and the collective output of the A-level neurons by s_A.

The output s_A of the A-layer serves as input to the neurons in the B-layer. Here each neuron B_j has two dendrites d_{j1} and d_{j2}. The dendrite d_{j1} has only one synaptic site on which only an axonal fiber of A_j terminates. The synaptic weight of this synapse is given by $b^{\ell}_{jj} = 0$, with $\ell = 0$. The second branch, d_{j2}, receives input from all the remaining neurons of the A-layer; i.e., from $\{A_1, \ldots, A_K\} \setminus \{A_j\}$. The synaptic weight of the synaptic site on d_{j2} for the terminal axonal fiber of neuron A_r, with $r \neq j$, is given by $b^{\ell}_{rj} = 0$, where $\ell = 1$. To compute the values $\tau^j_k(\mathbf{x})$ for the two dendrites of B_j, we use the general formula

$$\tau^j_k(\mathbf{x}) = p_{jk} \bigwedge_{i \in I(k)} \bigwedge_{\ell \in \mathcal{L}(i)} (-1)^{1-\ell} \left(x_i + w^{\ell}_{ijk}\right) \tag{8}$$

which is similar to eqn. 3. For $k = 1$ and $i = j$ we have $I(1) = \{1\}$ and $\mathcal{L}(j) = \{0\}$. Setting $p_{j1} = 1$ and employing eqn. 8 one obtains

$$\tau^j_1(s_A) = \bigwedge_{i \in I(1)} \bigwedge_{\ell \in \mathcal{L}(j)} (-1)^{1-\ell} \left(s^j_A + b^{\ell}_{jj}\right) = -s^j_A. \tag{9}$$

Similarly, for d_{j2} we have $k = 2$, $i = r$, $I(2) = \{1, \ldots, k\} \setminus \{j\}$, and $\mathcal{L}(r) = \{1\}$. Again setting $p_{j2} = 1$, one obtains

$$\tau^j_2(s_A) = \bigwedge_{r \in I(2)} \bigwedge_{\ell \in \mathcal{L}(r)} (-1)^{1-\ell} \left(s^r_A + b^{\ell}_{jr}\right) = \bigwedge_{r \neq j} s^r_A. \tag{10}$$

The values $\tau^j_1(s^j_A)$ and $\tau^j_2(s_A)$ flow into the cell body of B_j and its state is a function of the combined values received from its dendrites:

$$\tau^j_B(s_A) = \sum_{k=1}^{2} \tau^j_k(s_A) = \tau^j_1(s_A) + \tau^j_2(s_A) = \bigwedge_{r \neq j} s^r_A - s^j_A. \tag{11}$$

We consider the two possibilities of $\bigwedge_{r \neq j} s^r_A > s^j_A$ and $\bigwedge_{r \neq j} s^r_A \leq s^j_A$. The first possible case implies that $s^j_A \neq \infty$ and, hence, $s^j_A = d_1(\mathbf{x}, \mathbf{x}^j)$. That is, the pattern vector $\mathbf{x}$ is closer to the exemplar pattern $\mathbf{x}^j$ than any of the other exemplar pattern and within the allowable threshold T. The second possibility implies that either there is another exemplar $\mathbf{x}^r$ which is closer (or just as close) to $\mathbf{x}$ as $\mathbf{x}^j$, or that $\mathbf{x}^j$ surpassed the threshold T. In the first case we want the neuron B_j to send that information to the output neurons while in the second case we do not want B_j to fire. In order to achieve this we define the activation function to be the lattice-based hardlimiter

$$f_B(z) = \begin{cases} 0 & \text{if } z > 0 \\ -\infty & \text{if } z \leq 0 \end{cases}. \tag{12}$$

Thus, the output of B_j is given by $s^j_B = f_B[\tau^j_B(s_A)]$ and serves as the input to the output layer M. Each output neuron M_i, $i = 1, \ldots, m$, has only a single

dendrite d_{i1} receiving excitatory input from all K neurons of the B-layer. The weight associated with the synaptic site on d_{i1} of the terminal axonal fiber of B_j is defined as $w^1_{ji} = y^j_i$. Here $j = 1, \ldots, K$ and $i = 1, \ldots, m$. Using eqn. 3 to compute the output pattern, we note that since each M_i has only one dendrite d_{i1} we have $k = 1$ (for each i) and $I(1) = \{1, \ldots, , K\}$. Also, since we are dealing with excitatory synaptic responses only, we have that for each $j \in I(1)$, $\mathfrak{L}(j) = \{1\}$. By setting $p_{i1} = 1$, eqn. 3 now reduces to

$$\tau^i_1(s_B) = \bigvee_{j=1}^{K} (s^j_B + w^1_{ji}) = \bigvee_{j=1}^{K} (s^j_B + y^j_i). \tag{13}$$

Observe that $\tau^i(s_B) = \tau^i_1(s_B)$ The activation function for each neuron M_i is simply the identity function so that the output y_i of M_i is given by $y_i = \tau^i(s_B)$. The total output of the the set $M_1, \ldots, M_m$ is the vector $\mathbf{y} = (y_1, \ldots, y_m)$. It remains an easy excercise to show that for an uncorrupted input $\mathbf{x}^j$ the output at the M-level will be $\mathbf{y}^j$.

4 Experiments with Noisy and Corrupted Inputs

In this section we present results of some computational experiments that demonstrate the performance of the proposed DLAM in recalling stored associations when presented with corrupted versions of exemplar patterns. We use images to form pattern vectors only to provide a visual interpretation of the recall. In general. Associative memories are used for pattern recall, not image recall. The transformation of images into vectors is accomplished via the usual column-scan method. We created a database of image patterns from image obtained from various websites.

Experiment 1. In this experiment, each of the sets X and Y consists of six Boolean exemplar patterns. The set X is derived from the set of six 700×350 Boolean images shown in the top row of Figure 2, while the set of associated output patterns is derived from the six 380×500 Boolean images shown in the botton row of Figure 2. Thus, $X = \{\mathbf{x}^1, ..., \mathbf{x}^6\}$, with $x^j \in \{0, 1\}^{245000}$, and $Y = \{\mathbf{y}^1, ..., \mathbf{y}^6\}$ with $\mathbf{y}^j \in \{0, 1\}^{190000}$.

Every pattern image was corrupted adding "salt and pepper" noise. Each noisy pixel of corrupted image is rounded to either 0 or 1 to preserve the Boolean character of the images.

The range of the noise levels varied from 1% to 99% and was tested on all the images. Instances of corrupted input images are shown in Figure 3. The corresponding output images recalled by the DLAM are shown in the bottom row. The DLAM shows perfect recall robustness to salt and pepper noise.

Experiment 2. In this example we use a database of grayscale images in which the value of each pixel has an integer intensity value in a range from

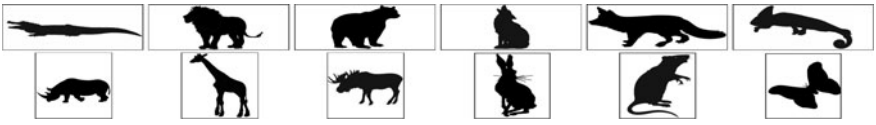


Fig. 2. Set of Boolean images of six predators in the first row and corresponding six preys in the second row

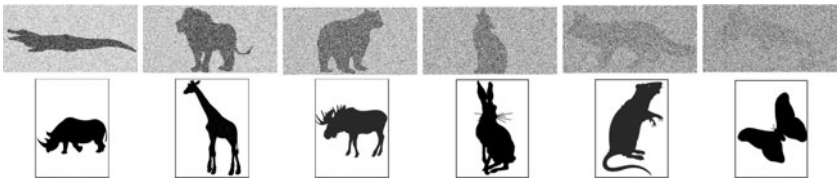


Fig. 3. First row: Boolean exemplar images corrupted with increasing levels of "salt and pepper" noise of 50%, 60%, 70%, 80%, 90%, and 94% (left to right). Bottom row: Perfect recall associations derived from the noisy input patterns in the top row.

0 (black) to 255 (white). Similar to Example 1, we use predator-prey association images as shown in Figure 4. Both predator and prey images are of size 265×265. In mathematical terminology we have $X = \{\mathbf{x}^1, ..., \mathbf{x}^K\} \subset \mathbb{R}^{70225}$ and $Y = \{\mathbf{y}^1, ..., \mathbf{y}^K\} \subset \mathbb{R}^{70225}$, $K = 5$. In this experiment we use different types of pattern corruption and noise. Specifically, we simulate noise pattern acquisition by increasing and decreasing image contrast, approximating linear camera motion, applying circular averaging filter, employing the morphological transforms of dilation and erosion with different structuring elements, and by using Gaussian and uniform noise. Figure 5 shows some of the tested image corruption changes. Different types of noise corruption have been applied to different images. The first column represents a motion blur, the 2nd Gaussian noise, the 3rd the application of a circular averaging filter, the 4th a morphological erosion with a line as structuring elements and the 5th a morphological dilation with elipsoid as structuring elements.

In the above two experiments, the threshold T for the activation function given in eqn. 7 was set to $T = \infty$; i.e, f_A was simply the identity function. With this threshold, the DLAM performance is very impressive in that associations can

Fig. 4. Set of grayscale images: 5 Predators in the first row and corresponding 5 Preys in the second row

Fig. 5. The exemplar input image patterns are shown in the 1st row. The 2nd through the 4th column below a given predator show the increase in the noise level or image corruption of the predator as discussed in the text. The bottom row illustrates the DLAMs recall performance when presented with a noisy predator image above the prey.

be recalled even at 99% random noise levels of the input data. However, images with such high and even lower noise levels of corruption cannot be identified by a human observer when not first shown the original pattern images. This poses the problem of misidentifying intruders. For example, suppose we let $\mathbf{x} \in \mathbb{R}^{70225}$ be obtained from a 265×265 image of a horse and present the DLAM with $\mathbf{x}$ as input. If $T = \infty$, then the DLAM will find the closest L^1-distance to one of the stored images and will associate the horse with one of the predators and correlate it with the predator's prey. To avoid intruders, a threshold $T < \infty$ can usually be determined that avoids misclassification of intruders. In image data (such as shown here) with random noise levels in excess of 60%, most images cannot be recognized by a human observer – the best visual pattern recognizer – when not first shown the corresponding non-noisy exemplar. Thus, if $\bar{x}^j$ represent exemplar x^j corrupted by about 60% of random noise, then setting $T_j = d_1\left(x^j, \bar{x}^j\right)$ and $T = \frac{1}{k}\sum_{j=1}^{k} d_1\left(x^j, \bar{x}^j\right)$ will, generally, present intruders be recognized as a legitimate exemplars. The next example supports this assumption.

Experiment 3. The dataset is the same as in Example 2. The recall of up to 99% of "salt and pepper" noise is perfect just as in Example 1. We consider the response of the DLAM to a new image pattern $\mathbf{x}$ which is not an element of X, namely the horse image of size 265×265 pixels shown in the last column of Figure 6.

Table 1. The distance ($\times 10^3$) between original predator image and the corrupted image with 50%, 60%, 63%, 65%, 70%, 80%, 90% and 100% of "salt and pepper" noise. The last column has the distance to the "horse" image shown in Figure 6.

Noise	0%	50%	60%	63%	65%	70%	80%	90%	100%	Horse
Leopard	0	4470	5374	5634	5813	6297	7158	8066	8932	**5667**
Eagle	0	4492	5348	5626	5844	6252	7154	8080	8947	6293
Wolf	0	4484	5396	5663	5832	6265	7177	8051	8965	6367
Dolphin	0	4452	5385	5640	5816	6281	7162	8059	8952	6713
Cobra	0	4487	5377	5621	5801	6292	7147	8052	8946	6189
Average	0	4477	5376	**5637**	5821	6277	7160	8062	8948	6246

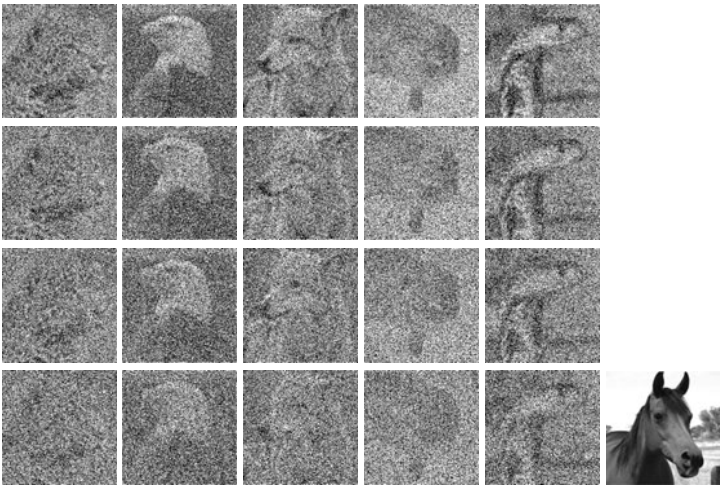


Fig. 6. Grayscale images from Experiment 3. The 1st, 2nd, and 3rd rows presents the input predator images corrupted with 50%, 60%, 63% "salt and pepper" noise. The 4th row contains corrupted images with the noise parameter set to 70%. These images are at the same distance from the original images as image "horse" in the last column.

If we present the image pattern x with the predator image that is closest (in the L^1-distance) to the horse and will, therefore, recall the prey associated with this predator. In this specific case the nearest predator is the leopard as can be ascertained from Table 1. Thus, the deer will be associated with the horse when the horse is used as input to the DLAM.

Note that a human observer will have extreme difficulty in identifying any of the images shown in Figure 6 if not shown the true exemplars first. Recognition at a noise level of 70% becomes pure guess work. Computing $T_j = d_1\left(x^j, \bar{x}^j\right)$ for

each j and each noise level as well as $d_1\left(x^j,\, x\right)$, we can see from Table 1 that $d_1\left(x^1,\, x\right) = 5667$, where $x^1 = leopard$ and $x = horse$, and $T = \frac{1}{5}\sum_{j=1}^{5} T^j = 5637$ when $\bar{x}^j$ represents as 63% corruption of x^j. Thus, T eliminates x as an intruder. Hence, using $T = 5376$ ($\bar{x}^j$ representing 60% corruption of x^j) would be an even better choice for preventing other intruders.

5 Conclusions

We present a new hetero-associative lattice memory based on dendritic computing. We report experimental results showing that this memory exhibits extreme robustness in the presence of various types of noise. It is our opinion that this DLAM is superior to existing hetero-associative memories. Further work will be addressed to perform exhaustive comparison tests with other associative memory architectures in order to rigorously verify our opinions.

References

1. Steinbuch, K.: Automat und Mensch, 2nd edn. Springer, Heidelberg (1963)
2. Steinbuch, K., Piske, U.A.W.: Learning Matrices and Their Applications. IEEE Trans. on Electronic Computers, 846–862 (1963)
3. Steinbuch, K.: Automat und Mensch, 3rd edn. Springer, Heidelberg (1965)
4. Steinbuch, K.: Automat und Mensch, 4th edn. Springer, Heidelberg (1972)
5. Kohonen, T.: Correlation Matrix Memory. IEEE Trans. on Computers C-21, 353–359 (1972)
6. Anderson, J.A.: A simple neural network generating an interactive memory. Mathematical Biosciences 14, 197–220 (1972)
7. Kohonen, T.: Self-Organization and Associative Memories, 2nd edn. Springer, Berlin (1987)
8. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. of the National Academy of Sciences, USA 79, 2554–2558 (1982)
9. Hopfield, J.J.: Neurons With Graded Response Have Collective Computational Properties Like Those of Two State Neurons. Proc. of the National Academy of Sciences, USA 81, 3088–3092 (1984)
10. Hopfield, J.J., Tank, D.W.: Computing with neural circuits. Science 233, 625–633 (1986)
11. Ritter, G.X., Sussner, P.: Associative Memories Based on Lattice Algebra. In: IEEE Inter. Conf. Systems, Man, and Cybernetics, Orlando, FL, pp. 3570–3575 (October 1997)
12. Ritter, G.X., Sussner, P., Diaz de Leon, D.L.: Morphological Associative Memories. IEEE Trans. on Neural Networks 9, 281–293 (1998)
13. Ritter, G.X., Diaz de Leon, D.L., Sussner, P.: Morphological Bidirectional Associative Memories. Neural Networks 12, 851–867 (1999)
14. Ritter, G.X., Urcid, G.: Lattice Algebra Approach to Single-Neuron Computation. IEEE Trans. on Neural Networks 14(2), 282–295 (2003)

15. Kaburlasos, V.G.: Towards a Unified Modeling and Knowledge Representation Based on Lattice Theory. Computational Inteligence 27(2006)
16. Kaburlasos, V.G., Ritter, G.X. (eds.): Computational Intelligence Based on Lattice Theory. SCI, vol. 67. Springer, Heidelberg (2007)
17. Ritter, G.X., Urcid, G.: Lattice Algebra Approach to Endmember Determination In Hyperspectral Imagery. In: Hawkes, P. (ed.) Advances in Imaging and Electron Physics, ch. 4, vol. 169, pp. 113–168. Elsevier, San Diego (2010)
18. Kaburlasos, V.G.: Granular Enhancement of Fuzzy-ART/SOM Neural Classifyers Based on Lattice Theory. In: Kaburlasos, V.G., Ritter, G.X. (eds.) Computational Intelligence based on Lattice Theory. SCI, vol. 67, pp. 3–23. Springer, Heidelberg (2007)
19. Graña, M., Villaverde, I., Moreno, R., Albizuri, F.X.: Convex Coordinates from Lattice Independent Sets of Visual Pattern Recognition. In: Kaburlasos, V.G., Ritter, G.X. (eds.). SCI, vol. 67, pp. 101–128. Springer, Heidelberg (2007)
20. Graña, M., Chyzhyk, D., García-Sebastián, M., Hernández, C.: Lattice Independent Component Analysis for functional Magnetic Resonance Imaging. Information Sciences 181, 1910–1928 (2011)
21. Chyzhyk, D., Graña, M.: Optimal Hyperbox Shrinking in Dendritic Computing Applied to Alzheimer's Disease Detection in MRI. In: Corchado, E., Snášel, V., Sedano, J., Hassanien, A.E., Calvo, J.L., Ślęzak, D. (eds.) SOCO 2011. AISC, vol. 87, pp. 543–550. Springer, Heidelberg (2011)
22. Chyzhyk, D., Graña, Savio, A., Maiora, J.: Hybrid Dendritic Computing with Kernel-LICA applied to Alzheimer's Disease detection in MRI. Neurocomputing 75(1), 72–77 (2012)
23. Ritter, G.X., Urcid, G.: Perfect Recovery from Noisy Input Patterns with a Dendritic Lattice Associative Memory. In: Proceedings of the International Joint Conference on Neural Networks (IEEE/INNS), San Jose, CA, pp. 503–510 (2011)
24. Urcid, G., Ritter, G.X., Valvdiviezo, J.C.N.: Grayscale Image Recall from Imperfect Inputs with a Two Layer Dendritic Lattice Associative Memory. In: Proceedings of IEEE, 3rd Congress on Nature and Biologically Inspired Computing, Salamanca, Spain, pp. 268–273 (2011)
25. Ritter, G.X., Urcid, G.: Learning in Lattice Neural Networks that Employ Dendritic Computing. In: Kaburlasos, V.G., Ritter, G.X. (eds.) Computational Intelligence Based on Lattice Theory. SCI, vol. 67, pp. 25–44. Springer, Heidelberg (2007)

Vascular Section Estimation in Medical Images Using Combined Feature Detection and Evolutionary Optimization

Iván Macía[1,2] and Manuel Graña[2]

[1] Vicomtech, Visual Communication Technologies
[2] Computational Intelligence Group, University of the Basque Country
imacia@vicomtech.org
http://www.ehu.es/ccwintco,
http://www.vicomtech.org

Abstract. Accurate detection and extraction of 3D vascular structures is a crucial step for many medical image applications that require vascular analysis. Vessel tracking algorithms iteratively follow vascular branches point by point, obtaining geometric descriptors, such as centerlines and sections of branches, that describe patient-specific vasculature. In order to obtain these descriptors, most approaches use specialized scaled vascular feature detectors. However, these detectors may fail due to the presence of nearby spurious structures, incorrect scale or parameter choice or other undesired effects, obtaining incorrect local sections which may lead to unrecoverable errors during the tracking procedure. We propose to combine this approach with an evolutionary optimization framework that use specific modified vascular detectors as cost functions in order to obtain accurate vascular sections when the direct detection approach fails. We demonstrate the validity of this new approach with experiments using real datasets. We also show that, for a family of medialness functions, the procedure can be performed at fixed small scales which is computationally efficient for local kernel-based estimators.

Keywords: Medical Image Analysis, Vascular Analysis, Vessels, Feature Detectors, Evolutionary Optimization, Vascular Tracking, Section Estimator, Medialness, Vesselness.

1 Introduction

Accurate detection and extraction of 3D vascular structures is a crucial step for many medical image applications that require vascular analysis [8] [7]. Vascular-related diseases, such as cerebrovascular accidents (stroke) or coronary artery disease, are caused by anomalies in the blood supply, like hemorrhages or blockages. Knowledge of patient-specific vascular structure is crucial for planning many interventions, such as neurointerventions or liver tumor resection. For these applications, many medical imaging modalities exist that are able to depict vessels. Among the most useful ones, we can mention X-ray Angiography,

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 503–513, 2012.

Computerized Tomography Angiography (CTA) and Magnetic Resonance Angiography (MRA), being the last two 3D modalities.

In order to obtain a meaningful and helpful vascular representation for quantification, visualization or other advanced analysis, it is first necessary to detect and extract the vascular structures with specialized image analysis methods.

Extraction procedures determine which points are part of the vascular structures, by using some measure of *vesselness* (likelihood of being part of a vessel) and geometric and/or appearance models of the vessels [4]. These algorithms usually obtain some geometrical and topological descriptors of the vascular network: the centerline of a vessel is often a good descriptor of the shape of the vessel along its path, and information about the local shape of the vessel is usually obtained by extracting sections along its centerline [8], and if possible by providing estimations of the local radius.

Tracking procedures [4] iteratively find the next vessel (center) point by advancing a given step size (fixed or adaptive) in the direction of the estimated local normal of the current vessel (center) point. However, these local normal and radius (and center point) estimators, which we will call *section estimators*, fail often due to the presence of nearby spurious structures, incorrect scale or parameter selection or other undesired effects. This may result in obtaining incorrect local sections which may lead to unrecoverable errors during the tracking procedure. On the other hand, estimating the correct local section may be interesting in other procedures other than tracking, for example for quantification of the section geometry, local curvature, centerline length, etc.

The present work focuses mainly on improving the accuracy and robustness of the section normal and radius estimation. We propose to combine the standard approach of obtaining a simple solution from the detector response with a non-linear evolutionary optimization procedure. The approach uses a 1+1 evolutionary strategy (ES) algorithm [12] for optimization and a cost function based on the classical section estimator approaches, in order to detect the local optimal orientation and size (radius) of the vascular structure. The optimization may also be useful to find the optimal parameters for the estimators.

The paper is organized as follows: Section 2 is a review of vascular detection and extraction procedures with focus on the methods used on the current work. Section 3 explains the procedure used to combine standard vessel feature detectors with an evolutionary optimization strategy. Section 4 describes some experiments on real datasets and corresponding results with the conclusions summarized on Section 5.

2 Review of Vascular Detection and Extraction

Vascular detection and extraction procedures have been widely reported in the literature. The proposed optimization procedure may be used with a high variety of detectors/estimators, due to its open nature. Here, we will focus on some of the most popular approaches that have been used in our implementation.

The detection procedure usually consists of obtaining a function or metric for every point of an image, called *vesselness function*, that expresses the likelihood of a pixel (voxel) of being part of a vessel structure. The design of such a function is based mainly on the basic property that vessels are usually visible as elongated hyperintense (or hypointense) structures on vascular images. If the vesselness value is higher in the centerline of vessels it is then called *medialness*.

The Hessian matrix is an important tool for vascular detection based on differential operators. For a three dimensional image $I : \mathbb{R}^3 \to \mathbb{R}$ the Hessian matrix $\mathcal{H}$ is defined as the matrix of (scaled) second order derivatives of the image

$$\mathcal{H}(\boldsymbol{x}, \sigma) = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \tag{1}$$

which describes the second order local image structure, that is, local image curvatures. The parameter σ is the scaling parameter and corresponds to the Gaussian smoothing, assuming that the derivatives are calculated in scale-space [5].

The three ordered eigenvalues λ_i, $\lambda_1 \leq \lambda_2 \leq \lambda_3$, of this Hessian matrix describe the principal image curvatures which best describe the local image second-order variations. The corresponding eigenvectors v_i describe the directions in which the principal curvatures occur. When the point $\boldsymbol{x}$ is close to the centerline or medial axis of a vessel and an appropriate scaling parameter is chosen, the local structure of the image is that of a bright (or dark) tubular structure, and the eigenvalues exhibit the following properties [10]:

$$\begin{aligned} &\lambda_1 \approx \lambda_2 \\ &\lambda_1,\, \lambda_2 \ll 0 \\ &\lambda_3 \approx 0 \end{aligned} \tag{2}$$

This assumes that the local curvature of the vessel is not too high and that the section shows radial symmetry. If these conditions are not met, the eigenvalues differ from this ideal situation.

The eigenvector $\boldsymbol{v}_3$ corresponds to the direction of the local vessel/tube axis where the curvature barely varies, hence λ_3 is almost zero. The other two principal curvatures, λ_1 and λ_2, occur in directions that go from the center of the tube to the external part of the vessel, where the curvature varies highly. Hence these eigenvalues are negative and of high absolute value (positive for a dark vessel in a bright background). The associated eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2$ are estimators of the local vessel section plane, since they are aligned with the directions of maximum curvature. Thus, they constitute a *section estimator* as described above.

Several detectors or filters may be designed using these second-order local structure properties. One approach is to take non-linear combination of the eigenvalues, trying to distinguish tube-like local structures from other shapes, such as plate-like or blob-like structures, which exhibit different relationships between the eigenvalues. For example, for plate-like structures two eigenvalues are similar to zero, and blob-like structures show three eigenvalues of the same relatively large value [2]. Of this kind are the methods of Sato *et al.* [10] and

Frangi *et al.* [2] among others. Other approaches estimate the vessel section using the obtained eigenvectors, and then use some sort of differential or integral operator. Examples are the offset medialness measure used by Krissian *et al.* [3] or the ridge detection approach used by Aylward *et al.* [1]. Most of these authors adopt multi-scale approaches, which first select a discrete range of scales, obtain responses for each scale, and then integrate them into a multi-scale representation, usually by taking the maxima across scales. This requires a normalization of derivatives across scales [5]. An estimate of the radius may be obtained by multiplying the scale which gives the maximum vesselness value by a factor which depends on the vessel intensity distribution [3].

The *offset medialness* measure [3] is an integral measure defined in the section plane as:

$$R_\sigma^+(\boldsymbol{x}, r) = \frac{1}{2\pi} \int_{\alpha=0}^{2\pi} -\nabla I_\sigma \left(\boldsymbol{x} + r\boldsymbol{v}_{\alpha_i}\right) \cdot \boldsymbol{v}_{\alpha_i} d\alpha \tag{3}$$

where $\boldsymbol{v}_\alpha$ is a rotating vector, or phasor given by

$$\boldsymbol{v}_\alpha = \boldsymbol{v}_1 \cos\alpha + \boldsymbol{v}_2 \sin\alpha \tag{4}$$

Equation 3 is the integral of the projection of the negate of the gradient vector in the radial direction of a circle of radius r around the considered point. This circle is located in the estimated section plane formed by eigenvectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. In fact, any other section estimator could be used. As we can see, by tuning r we have an estimate of the local vessel radius.

The corresponding discrete implementation samples the circle points in which the gradient is calculated and corresponds to:

$$R_\sigma^+(\boldsymbol{x}, r) = \frac{1}{N} \sum_{i=0}^{N-1} -\nabla I_\sigma \left(\boldsymbol{x} + r\boldsymbol{v}_{\alpha_i}\right) \cdot \boldsymbol{v}_{\alpha_i}, \qquad \alpha = 2\pi i/N \tag{5}$$

Pock *et al.* [9] use the gradient magnitude instead of the gradient projection. We believe that is better to use the projection in the radial direction determined by $\boldsymbol{v}_\alpha$ rather than the gradient magnitude, since spurious or adjacent structures may have a greater undesired contribution in terms of gradient magnitude, which may lead to large values of medialness where it should not. On the other hand, as an improvement, they introduce the following symmetry coefficient:

$$\omega(b_i) = \exp\left[-\frac{1}{2\xi^2}\left(1 - \frac{b_i}{R_\sigma^+}\right)^2\right], \quad \xi \in (0, 1] \subset \mathbb{R} \tag{6}$$

where

$$b_i = -\nabla I_\sigma \left(\boldsymbol{x} + r\boldsymbol{v}_{\alpha_i}\right) \cdot \boldsymbol{v}_{\alpha_i} \tag{7}$$

is the contribution of each radial point, also called *boundariness* [13]. Here, we have used the boundariness measure of Krissian *et al.* [3] but other boundariness measures could be used. The resulting adaptive medialness function is:

$$R_\sigma(\boldsymbol{x}, r) = \frac{1}{N} \sum_{i=0}^{N-1} \omega(b_i)\, b_i \tag{8}$$

The symmetry coefficient ξ penalizes asymmetry in the radial distribution of gradient values. When $\xi = 1$ no penalization is performed. The lower the value the more the asymmetry is penalized. There is a trade-off between the asymmetry of the section and the detection rate. If very asymmetrical sections are expected, the value should be one or close to one. Otherwise, $\xi = 0.5$ gives good results in most situations. We also divide the resulting medialness by one plus the gradient magnitude at the center point, since it should be low in a centerline point. This last step was also used by Pock *et al.* [9] but they substracted this value instead of dividing it.

The original implementation of Krissian *et al.* [3] makes the radius r dependent on the scale in the form $r = \tau\sigma$. In practice, it is not necessary to change r linearly with the scale. Additionally, with large diameters, we would need large scales with increased computational costs. A better approach is to choose a single or a few scales valid enough for the range of diameters to be considered and then adjust r to obtain a maximum response.

Next, we explain our hybrid method of combining this offset medialness measure with an optimization procedure so as to obtain an optimal section estimator.

3 Vascular Feature Detection with Evolutionary Optimization

The vascular feature detection with evolutionary optimization procedure consists of converting a vesselness measure into a cost function that is optimized with respect to a set of parameters. Currently, we use the optimization in order to obtain an optimal section estimator. For this purpose, the vesselness measure needs to be a medialness measure, with the largest values on the vessel axis. In our experiments, we have used the offset medialness measure in eq. 8. The problem can be expressed mathematically as:

$$\arg\max_{\boldsymbol{u}\in\Omega} R_\sigma(\boldsymbol{x}_c, \boldsymbol{u}), \quad \Omega = \left\{\boldsymbol{u} = (\boldsymbol{n}, r) \in \mathbb{R}^4\right\} \quad \text{s.t. } \|\boldsymbol{n}\| = 1 \tag{9}$$

The optimization procedure tries to find the optimal unit section normal $\boldsymbol{n}$ and radius r of the medialness at each section center point $\boldsymbol{x}_c$(assuming that is the real vessel section center). This would involve a 4D parameter space for optimization. However, the components of the unit section normal, which are the director cosines, are related to each other by the expression:

$$\|\boldsymbol{n}\| = \sqrt{n_x^2 + n_y^2 + n_z^2} = 1 \tag{10}$$

Then, the optimization procedure can be expressed as:

$$\arg\max_{\boldsymbol{u}\in\Omega} R_\sigma(\boldsymbol{x}_c, \boldsymbol{u}), \quad \Omega = \left\{\boldsymbol{u} = (n_x, n_y, r) \in \mathbb{R}^3\right\} \quad \text{s.t. } \{|n_x| < 1, |n_y| < 1\} \tag{11}$$

This means that we have a 3D search space with two unit normal coordinates and the radius of the detector since the last coordinate is calculated with the above formula[1]. The new constraints for the n_x and n_y coordinates can be implemented very easily by returning a zero value for the cost function when the constraints are not met. This is a fast and simple alternative to other more complex approaches such as using Lagrange multipliers.

Note that here the section center is not optimized and it is assumed to be previously calculated, but it could be incorporated into the procedure. The scale σ of the derivative calculations could also be included into the optimization. However, gaussian scale-space derivatives are calculated locally using an implementation with discrete kernels [6] and this would require the calculation of a large kernel at each optimization step.

The procedure for obtaining the section normal then becomes a two stage method (see Figure 1), assuming that we are located on a vessel center point:

1. Estimate the local section using a standard non-optimized estimator. This gives a single solution for the section normal, given the scale, radius and center point. The initial parameters are chosen from the neighbor point if previously calculated. A multiscale approach tests a discrete range of scales and selects the scale that yields the maximum medialness value.
2. Compute the best parameters for the optimization problem in eq. 11 using a (1+1)-ES evolutionary optimizer. Take as starting point the parameters and value of the section normal and radius calculated on the first stage.

Next, we proceed to describe our experiments with real datasets.

4 Experiments and Results

We tested our optimization methods with real 3D datasets, one Contrast-enhanced Magnetic Resonance Image (MRI) of the liver, one Magnetic Resonance Angiography (MRA) of the abdomen and one Computerized Tomography Angiography (CTA) of the abdomen. The resolution of the data was variable, with the liver MRI 1.56x1.56x3.0 mm. spatial resolution, the CTA with 0.72x0.72x1.5 mm. and the MRA 1x1x1.5 mm.

For each dataset, we manually delineated the approximate centerline of one or two long vessels: one major liver vein in the MRI dataset, the aorta in the MRA dataset, and iliac arteries in the other two CTA datasets. The points were interpolated by a B-Spline curve which was then sampled in order to increase the number of centerline points.

[1] Note that this would not be true with standard variable-length vectors.

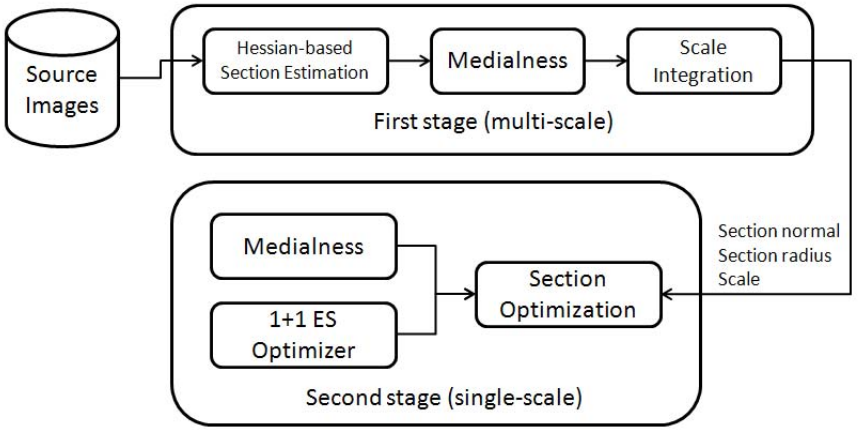


Fig. 1. Two-stage vessel estimation scheme used in our experiments

First, we estimated the sections by the direct method of calculating the eigenvectors of the Hessian matrix. In order to select the scale, for each centerline point, we computed the offset medialness in eq. 8 in the estimated section plane and chose the parameters for the best value (scale, section normal and radius). We used a discrete range of scales ranging from 1.0 to 7.0 using a step size of 1.0. The radius used was the scale times a factor of $\sqrt{3}$ which is a good radius estimate for Gaussian tubes [3]. For all our experiments we used $\xi = 0.5$ for the medialness asymmetry parameter.

Second, we computed the sections with our optimization scheme. In order to keep the two normal components in the range $[-1, 1]$, we simply returned zero as the medialness value outside this interval. The radius was also constrained in the range $[0, R_{max}]$ where R_{max} is chosen above the maximum expected radius value on the images. The scale was fixed in all our experiments to $\sigma = 1.0$, since we found out that the detection was more sensitive to the radius.

The optimization scheme used a non-linear optimization algorithm called (1+1)-Evolution Strategy (ES) [12] as implemented in [14], which belongs to the family of Evolutionary Algorithms [11]. As initial parameters, we chose the normal and the radius from the first step. The medialness was calculated each time on the estimated section. The stop condition was either 5000 iterations or a minimal search radius of 0.25 (Frobenius norm of the covariance matrix). Most of the times the procedure was finished after about 2000 iterations. Note, that our focus here was to test the validity of the approach and not the performance of the optimizer. The latter has quite a lot of margin for improvements, for example, by trying to reduce the search space or by tuning the parameters for optimal performance.

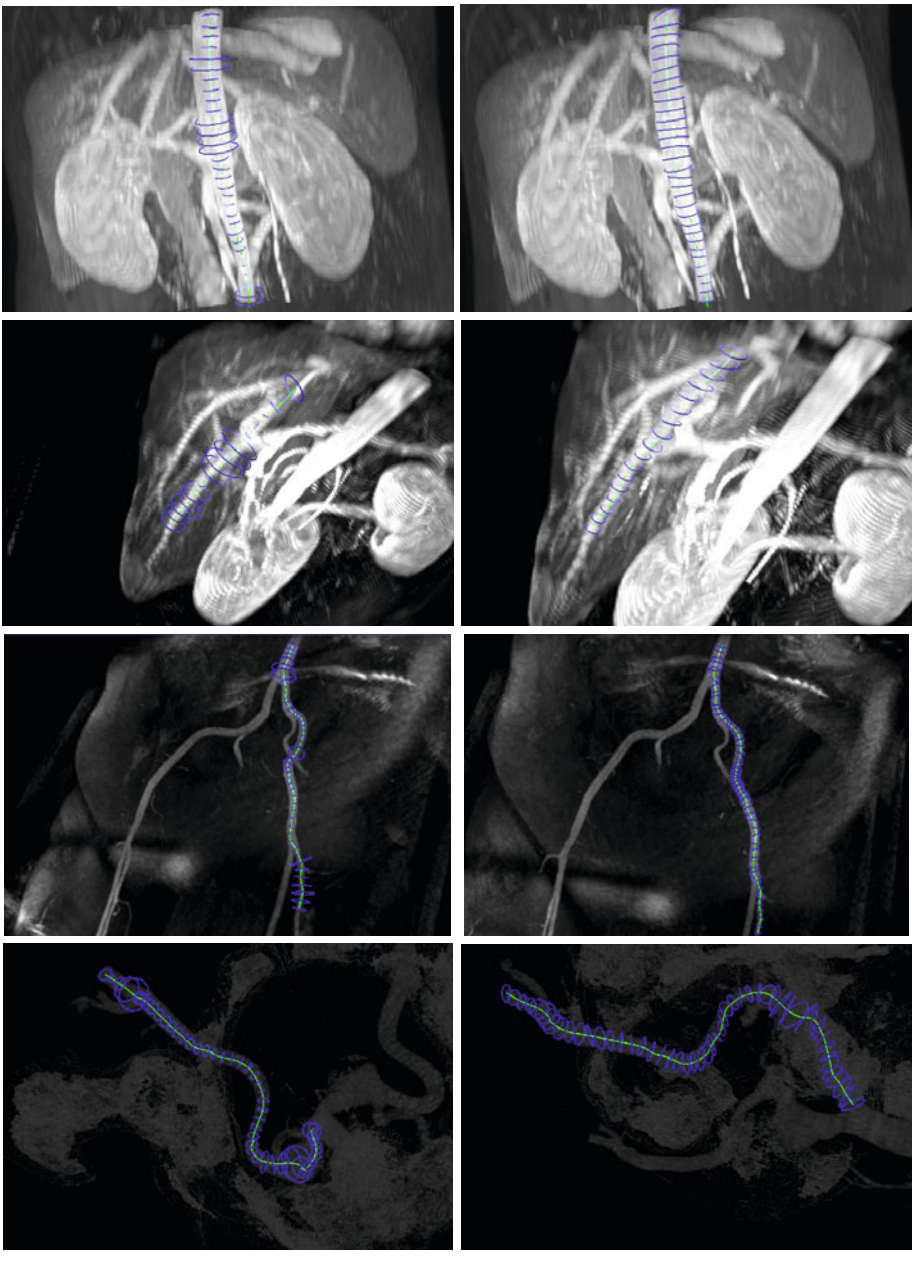


Fig. 2. Volume rendering of real datasets with rendering of estimated vessel sections. Delineated centerlines are shown in green and estimated sections in blue. For each row, from top to bottom, results for an aorta in a MRI, one major liver vessel in the same MRI, and iliac arteries for a MRA (third row) and CTA (fourth row) study. Left column depicts the results of the first, direction estimation stage. Right column shows the results after the evolutionary optimization procedure.

Results of the described method for both stages are shown in Fig. 2. The 3D render shows the estimated sections and radius depicted as circles at each centerline point (actually we did not draw all the centerline points but only a subset). Note that the standard estimator works quite well at estimating the sections. This is normal since most of the vessels were clearly visible. However, there was a high variation in the scale and radius estimation along the vessels. The optimized procedure shows very precise results at estimating the section and radius, except maybe at bifurcations, where the first stage also fails. Note specially that the accuracy in the radius estimation is really high, which would be difficult to estimate by manually setting the parameter on the first stage.

It is important to highlight the fact that our method can be applied to virtually any vesselness function. In this sense, the method can be thought of as both a shape and parameter estimator, thus decreasing the number of parameters of the original estimator. In our experiments, we have initialized the parameters for each section independently of the results of the previous optimization. However, the optimizer can be initialized with an initial position corresponding to the previously calculated point. In this way, the optimization procedure would be less time consuming.

The optimization stage is slower than the previous step (in the order of minutes, rather than in the order of seconds). In practice, it should only be used when we accurate values of radius and section normal are required or when the value of the direct section estimator is likely to be incorrect. During a tracking procedure, this can be detected as an outlier, for example, when the normal exceeds an angle with respect the previous normal along the vessel path (assuming that the step size is small enough). It may also be used as a parameter estimator for the standard procedure obtaining parameter values to be used in a given application.

On the other hand, the scale for the medialness was fixed to a small value in the optimization stage. The reason is that the scale for this family of medialness function should be chosen according to the size of the vessel boundaries (the boundary is relatively thin) and not according to the diameter. Otherwise, precision would also be penalized, since we would have a poorer localization with higher scales. This is an important conclusion, since we have observed that, for these types of vesselness functions, we can operate at lower scales and with less variability. The reason is that the scale of the diameters may vary considerably but the scale of the vessel boundaries not so much. For local calculations using discrete kernels, this supposes smaller kernels and less kernel recalculations, which is computationally faster. The procedure also does not require estimating the Hessian at each iteration, which makes it faster than expected.

5 Conclusions and Future Work

We have developed a method for the estimation of vessel sections on medical images. It uses an evolutionary optimization scheme together with well-known vascular feature detectors. These were adapted as cost functions and acted as

section estimators for the section normal and radius. This alternative approach is used after a standard direct multiscale section estimator stage. In the current work, we have used a family of medialness functions as section estimators, although the method admits other types of estimators. We have tested the validity of the approach by estimating the vessel sections of delineated vessel centerlines on real MRI, MRA and CTA datasets. Our results show improved accuracy, more evident in the radius estimation, at the expense of extra computational time. The procedure can be used as a high accuracy estimator, as a backup stage during a tracking procedure or as a parameter estimation for several vessel feature detectors.

Future work will be focused on more exhaustive experimental work, extending the approach to use other types of section estimators and improving the performance of the optimization procedure.

References

1. Aylward, S.R., Bullitt, E.: Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. IEEE Trans. Med. Imaging 21(2), 61–75 (2002)
2. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale Vessel Enhancement Filtering. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 130–137. Springer, Heidelberg (1998)
3. Krissian, K., Malandain, G., Ayache, N., Vaillant, R., Trousset, Y.: Model based detection of tubular structures in 3d images. Computer Vision and Image Understanding 80(2), 130–171 (2000)
4. Lesage, D., Angelini, E., Bloch, I., Funka-Lea, G.: A review of 3d vessel lumen segmentation techniques: Models, features and extraction schemes. Medical Image Analysis 13(6), 819–845 (2009)
5. Lindeberg, T.: Discrete derivative approximations with scale-space properties: A basis for low-level feature extraction. J. Math. Imaging Vision 3, 349–376 (1993)
6. Macía, I.: Generalized computation of gaussian derivatives using itk. The Insight Journal (December 2007)
7. Macía, I., Graña, M., Maiora, J., Paloc, C., de Blas, M.: Detection of type ii endoleaks in abdominal aortic aneurysms after endovascular repair. Computers in Medicine and Biology 41(10), 871–889 (2011)
8. Macía, I., Graña, M., Paloc, C.: Knowledge management in image-based analysis of blood vessel structures. Knowledge and Information Systems 30(2), 457–491 (2012)
9. Pock, T., Janko, C., Beichel, R., Bischof, H.: Multiscale medialness for robust segmentation of 3d tubular structures. In: 10th Computer Vision Winter Workshop (2005)
10. Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., Kikinis, R.: 3d multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. Medical Image Analysis 2(2), 143–168 (1998)

11. Schwefel, H.P.: Evolution and Optimum Seeking. Wiley (1995)
12. Styner, M., Brechbuhler, C., Székely, G., Gerig, G.: Parametric estimate of intensity inhomogeneities applied to mri. Trans. Med. Imag 19(3), 153–165 (2000)
13. Xu, M., Pycock, D.: A scale-space medialness transform based on boundary concordance voting. J. of Math. Imag. and Vision 11(3), 277–299 (1999)
14. Yoo, T.S., Ackerman, M., Lorensen, W., Schroeder, W., Chalana, V., Aylward, S., Metaxas, D., Whitaker, R.: Engineering and algorithm design for an image processing api: a technical report on itk - the insight toolkit. Stud. Health. Technol. Inform. 85, 586–592 (2002)

Modifications of Classification Strategies in Rule Set Based Bagging for Imbalanced Data

Krystyna Napierala and Jerzy Stefanowski

Institute of Computing Science, Poznań University of Technology,
60–965 Poznań, Poland
{krystyna.napierala,jerzy.stefanowski}@cs.put.poznan.pl

Abstract. Learning bagging ensembles of rule classifiers from imbalanced data is considered. We claim that simply introducing bagging instead of single classifiers may not bring the expected improvement in recognizing a minority class. The reason lies in the classification strategies of component classifiers, which are biased toward majority classes when no-matching or multiple-matching conflicts between rules occur. We argue that abstaining, i.e. allowing component classifiers to refrain from giving a prediction in ambiguous situations, may help to correctly recognize minority examples. Our evaluation on 17 imbalanced datasets and 5 classification strategies shows that bagging with abstaining is better than both standard bagging and single rule based classifiers.

Keywords: Bagging, Abstaining Classifiers, Imbalanced data.

1 Introduction

This paper studies the classification strategies of component classifiers inside bagging ensembles and their influence on the recognition of the minority class in imbalanced data. Learning classifiers from imbalanced data has become a research challenge in data mining [9]. A data set is considered to be imbalanced if one of target classes contains much smaller number of examples than the other classes. The popular learning algorithms do not work properly on such data as they are biased toward the majority classes and fail to correctly classify examples from the minority class. In this study we concentrate on improving ensembles of rule-based classifiers, as they are strongly affected by class imbalance problem.

Several specialized methods have already been proposed to improve classifiers learned from imbalanced data – for their review see e.g. [9]. They are categorized in two groups. The first group includes classifier-independent methods that transform the original data to change the distribution of classes, e.g. by re-sampling. The other group involves modifications of a learning phase, classification strategy, ensembles or adaptation of cost sensitive learning. Most of the research with ensembles concerns modification of boosting, for instance by incorporating additional sampling or other schemas for re-weighting of examples [7,9]. Not many proposals refer to adapting bagging for class imbalance and they mainly change a sampling step [7].

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 514–525, 2012.

In our paper we consider another possibility of modifying bagging which operates during the aggregation of votes coming from the component classifiers represented by sets of rules. While using an *unordered set* of induced rules to classify new coming examples, conflict situations may occur when the description of a new example *matches* (satisfies) many rules from different classes or when it does not match any rule. Single rulesets-based classifiers are usually designed to always assign a class label for a new example, using specialized *classification strategies* for solving conflicts between unordered rules. However, in case of an ensemble another point of view is also possible.

Stefanowski et al. hypothesized in [2] that a component classifier in the ensemble may refrain from making a prediction when no rule covers a classified example. Motivation behind it is that each set of unordered rules covers a subspace in the problem space which can be seen as an area of its "expertise". Thus, it is likely that if one classifier abstains from classifying an example, other more competent sets of rules should participate in voting about its class label. Preliminary results showed that such an idea improved the overall accuracy for bagging ensemble [2] applied with MODLEM rule induction algorithm [11] and Grzymala's classification strategy [8]. An analysis of Grzymala's and other popular classification strategies has led us to conclusion that they are strongly biased toward the majority class when no-matching or multiple-matching occurs. Thus, we claim that introducing abstaining may be effective for improving recognition of the minority class, as it may prevent classifiers incorrectly indicating the majority class from outvoting other classifiers in the ensemble pointing to the minority class.

Therefore, the main aim of our study is to analyse the influence of introducing abstaining in component rule-based classifiers in bagging on the recognition of the minority class in imbalanced data. As a basic algorithm inducing unordered sets of rules we chose MODLEM due to the previous experience of one of the authors in various ensembles [11]. Moreover, it has already been used in class imbalance problems, see e.g. [12]. Comparing to the previous research, we will also study more rule classification strategies based on different principles.

2 Related Works

We briefly describe only these ensemble methods which are the most related to our proposal. Firstly, popular multiple classifier systems which increase the classification accuracy are biased toward the majority classes and have to be modified to improve recognizing the minority class. A comprehensive review of addressing class imbalance by modified ensembles has recently been published in [7]. According to their taxonomy, the existing proposals can be divided into *combining pre-processing methods* with ensembles and *cost-sensitive boosting* approaches. Most of the proposals concern modifications of boosting. Among the first group, re-weighting phase in each iteration is often integrated with re-sampling techniques, to train the next classifier more toward the minority class. The most well known proposal is SMOTEBoost, in which the learning sets created in each

iteration of AdaBoost are pre-processed by the SMOTE method, i.e. special oversampling of the minority class with synthetic examples. DataBoost-IM method combines a different technique for generating synthetic examples with AdaBoost. See the descriptions of both methods in [7,9].

In case of bagging there are only few modifications. They work in the first phase of learning to change the distribution of examples in bootstrap samples. *Overbagging* methods oversample the minority examples in each bootstrap, e.g. using SMOTE, while *underbagging* methods apply an undersampling of the majority examples [7].

Cost-sensitive boosting methods introduce cost items into the weight update formula, which treat the examples differently depending on their class. The idea is to give heigher weights to the misclassified minority examples. The reader is referred to a review in [9] where two modifications of AdaBoost – AdaC1 and AdaCost – are described.

The idea of *abstaining* – i.e. refraining from class predictions – has already been considered, in particular in cases when the classification is uncertain. It may occur when a classified example is located either in the boundary between the classes or very far from any class. Some techniques, such as threshold classifiers (e.g. neural networks or Bayesian), which produce a distribution of probabilities of memberships to different classes, may naturally abstain from classification if none of the probabilities exceeds a preferred threshold.

Such classifiers have also been studied in the framework of ensembles. However, most of the research concerns refraining from the final decision in case of disagreement between votes of component classifiers – see e.g. [10] shows that it may improve the final accuracy.

In our proposal, single component classifiers are allowed to give no answer. To some extent, other rule ensembles like SLIPPER [6] in which component classifiers are *single rules*, incorporate quite similar idea – a rule is excluded from voting if a new example is not covered by it. However, to the best of our knowledge, there are no abstaining solutions for ensembles where component classifiers are *sets of rules*.

3 Rule Induction and Classification Strategies

3.1 MODLEM Algorithm

To induce rules we have chosen a MODLEM algorithm, due to the reasons explained in the introduction. It has been introduced by Stefanowski in the 90's; see also its detailed description in [11]. Here we only briefly present its main characteristics. It is based on the idea of *sequential covering* and it iteratively generates an *unordered minimal set* of rules for every target concept (class) which is represented by its positive examples. The main procedure for rule induction starts from creating a rule by sequentially choosing the best elementary conditions according to some criteria (in our experiments we used an entropy-based one). When the rule is stored, all positive learning examples covered by this rule are removed from consideration. The process of looking for the next

rule is repeated while some significant positive examples remain uncovered. For inconsistent data, rule pruning or rough sets approximations can be used. An additional specificity of the MODLEM algorithm is its good ability to handle directly numerical attributes during rule induction when the elementary conditions of rules are created, without any preliminary discretization phase.

MODLEM was successfully used within the framework of multiple classifiers, including also bagging – for a review see [11]. We also studied it integrated with different informed re-sampling techniques which deal with class imbalance [12].

3.2 Classification Strategies

Prediction of a class label is based on *matching* an attribute description of a new coming example to the condition parts of induced rules. The classification strategies are necessary for solving possible *conflict situations*, e.g. when the new example's description satisfies conditions in rules from opposite classes or when it does not match any rule. The strategies can be divided into two groups depending whether the rules are ordered or not. In the first case, rules are arranged into the priority *ordered list* and a new example is classified according to the first completely matched rule. For handling non-matching, a special *default rule* is added at the end of the list and it usually indicates the majority class. In case of an *unordered set of rules*, the new example's description is tried against each rule in the set. Conflict situations are now: *multiple matching* to several rules from opposite classes and *no-matching*. As MODLEM induces an unordered set of rules, we briefly describe the selected classification strategies.

Grzymala's LERS Strategy
We chose this strategy as it is most often used with MODLEM. It was originally introduced by Grzymala in [8]. It is based on a *voting* of matched rules according to their supports $sup(r)$ (i.e. number of learning examples covered by the rule). The total *support* for a class K is defined as: $sup(K) = \sum_i^m sup(r_i)$, where r_i is a matched rule that indicates K, m is the number of these rules. A new example is classified to the class with the highest total support. In case of no-matching, so called *partial matching* is considered where at least one of rule conditions is satisfied by the corresponding attributes in the new example's description x. The matching factor *match(r,x)* is introduced as a ratio of conditions matched by the object x to all conditions in the rule r. The total support is modified to

$$sup(K) = \sum_i^p match(r, x) \times sup(r_i) \tag{1}$$

where p is the number of partially-matched rules.

Nearest Rules Strategy
This strategy, introduced by Stefanowski, is also based on the idea of voting with rule supports. However, a rule support is calculated in a different way – if the rule covers examples from different classes, then their numbers are included in the total supports for each class K. Then, the main difference lies in solving a no-matching case by means of so-called *nearest rules* instead of partially

matched ones [11]. These are rules nearest to the object description with respect to the *valued heterogeneous metric* HVDM. It aggregates normalized Euclidean distances for numeric attributes with Stanfil and Valtz value difference metric for nominal attributes [13]. In case of numeric attribute, the difference between an attribute value in the example's description and an appropriate rule condition is calculated to the nearest threshold value in the rule condition. A coefficient expressing rule similarity (complement of the calculated distance) is used instead of matching factor in the Equation 1 and again the strongest decision class K wins. While computing this formula *only the first k nearest rules* are considered to reduce bias for the majority class rules.

Default Rule Strategy
In this strategy a default rule is used in case of no-matching, which assigns the example to the majority class. In case of multiple matching, it uses the same solution as Grzymala's strategy. We chose this strategy, because it is often used in the algorithms inducing unordered sets of rules, e.g. in a version of CN2 [4].

Discrimination Measure Strategy
As an alternative strategy which uses a different rule quality measure, we apply a proposal of Aijun An [1]. Unlike rule support, its rule quality measure is defined as a *measure of discrimination* $DM(r) = log\frac{P(r|K)\times(1-P(r|\neg K)}{P(r|\neg K)\times(1-P(r|K))}$, where P denotes probability and r refers to a rule. For more technical details of estimating probabilities and adjusting this formula to prevent zero division see [1]. Its interpretation says that it measures the extent to which rule r discriminates between positive and negative examples of class K. Inside the classification strategies it is used in similar formulas for decision scores as in the Grzymala's strategy - the only difference concerns putting $DM(r)$ in place of $sup(r)$. As supports implicitly take into account the class cardinality and discrimination measure uses probabilities which are independent of class cardinalities, DM strategy may be less biased towards majority classes than Grzymala's strategy.

M-Estimate Strategy
This strategy is the best representative of approaches that solve conflict situations by choosing a single rule according to its quality measure. Generally, this measure reflects the accuracy of a rule when classifying new examples. Several researchers proposed to use the *m–estimate* [5] of rule probability. For a rule r indicating class K, it is defined as: $m(r) = \frac{n^K(r)+m\times P_a(K)}{n(r)+m}$, where $n(r)$ is a number of learning examples covered by rule r, $n^K(r)$ is a number of these examples which belong to K, $P_a(K)$ is an *a priori probability* of this class (estimated on the learning set). The m parameter allows to tune this estimate with respect to the data and problem characteristics (usually more noise requires larger m). In our experiments, m will be set to 2 according to [5].

The multiple matching is solved by selecting one rule in the conflict set with the highest value of the m-estimate. However, in case of no-matching a default rule is usually applied. As default rule means classifying to the majority class, we decided to reduce this bias by introducing the idea of partially matched rules.

More precisely, for each of these rules we calculate the m-estimate for the rule with the condition part reduced to matched conditions only. Then, we make a classification decision according to one partially matched rule with the highest m-estimate.

4 Abstaining in Bagging

The concept of abstaining in ensembles can be considered at two levels: either it may occur in the final decision of the ensemble or the component classifiers may refrain from prediction. The second option is the topic of our research.

Let us notice that each rule locally covers some learning examples. Then, a set of rules covers a part of learning examples being a sum of local covers – this is a different situation to, e.g., decision trees which globally divide the space of attributes describing learning examples. Such a limited cover of the original data is even more visible if the rule classifiers are induced from the bootstrap samples inside bagging.

Single classifiers are designed to always indicate a class label. However, we claim that abstaining is more reasonable for rule ensembles. Each set of rules inside the component classifier covers a certain part of learning examples, which can be treated as its area of expertise. These areas can overlap for different classifiers in bagging. However, due to the diversity of component classifiers (obtained by bootstrap sampling [3]) one should rather expect not too much overlapping. We claim that if a new example does not match any rule in the component classifier, this classifier should refrain from predicting its class label as it has insufficient information about this example. In other terms, the example is located in the part of attribute space outside the area of this classifier's competence. Other classifiers, which are more likely to be competent for the classified example, should take part in making a final decision of the ensemble.

In our proposal we decided to modify the bagging ensemble [3]. The first part of its construction is unchanged, i.e. we use several bootstrap samples from the original learning set (uniform sampling with replacement) and then generate the sets of rules by the unchanged algorithm. However, we modify a phase of producing predictions by component classifiers. If a classified example does not match any rule, the component classifier is *switched off* from taking part in the final decision. The predictions of the remaining classifiers are aggregated with the majority equal weight voting scheme as it is done in standard bagging.

This idea was introduced by Stefanowski et al in [2]. However, it was considered only with respect to the overall classification accuracy. Experiments with the component rule sets classifiers, induced by MODLEM and used with LERS strategy, showed that abstaining improved the classification accuracy in bagging and in adaptive Ivotes ensemble [2]. A positive influence was more visible for bagging. However, we have not tested this property with respect to class imbalance.

Let us remind that typical classification strategies are biased toward better recognizing the majority classes than the minority one. Nearly all strategies described in Section 3 rely on voting with rule evaluation measures to solve conflicting situations (i.e., multiple match and no match). Such a voting scheme, e.g. with rule supports, is biased toward the majority classes, as rules from these classes are stronger (supported by a larger number of learning examples) and more general than rules for the minority class. Thus, without changing the classification strategy a minority class is likely to be outvoted when no match or multiple match occurs, resulting in incorrect predictions. Some solutions have already been proposed for single classifiers (e.g. strengthening the minority rules). However, yet another solution, in our opinion suitable for bagging ensembles, is just to abstain in case of no-match conflicts.

5 Experiments

We plan to evaluate the influence of introducing abstaining of component classifiers on the ability of the modified bagging to recognize the minority class. We will compare the modified bagging with abstaining with a single rule set classifier and with a classical bagging. These two classifiers will give a reference baseline for studying the impact of our modification. Moreover, in case of a single classifier we want to compare the selected classification strategies to check whether some of them can better deal with a minority class. Component classifiers and a single classifier will be induced by MODLEM algorithm.

All the classifiers are based on unprunned sets of rules as it can help to recognize the minority class. A number of classifiers in bagging ensemble is 30 because in the earlier studies [2] we have observed that it lead to better total accuracy without increasing too much the computational costs. Moreover, we have observed that on average 3-4 components may abstain, so the number of classifiers cannot be too small. All implementations are based on the WEKA toolkit[1] and our own implementation of MODLEM[2].

As improving the recognition of the minority class may decrease the accuracy for the majority class, we evaluate the classifier performance with respect to *sensitivity* (true positive rate or, in other words, accuracy of the minority class) and *specificity* (accuracy of the majority class); see [9] for their detailed definitions and justification. To have a more balanced evaluation by a single measure, as a next criterion we chose a G-mean measure – a geometric mean of sensitivity and specificity.

All experiments are carried out on 17 data sets from the UCI repository[3] except *acl*[4]. We selected the datasets that are characterized by varying degrees

[1] www.cs.waikato.ac.nz/ml/weka

[2] Some classification strategies were implemented by our M.Sc. student Szymon Wojciechowski. We are grateful for his help.

[3] http://www.ics.uci.edu/~mlearn/MLRepository.html

[4] Acl data was collected by prof. K. Slowinski and dr D. Siwinski from Poznan University of Medical Science.

Table 1. Description of datasets

Dataset	Abbrev.	Number of examples	Imbalance ratio [%]	Minority class
acl	AC	140	28	1
breast-cancer	BC	286	29	rec-events
bupa	BU	345	42	sick
car	CA	1728	4	good
cleveland	CL	303	12	positive
cmc	CM	1473	22	long-term
credit-g	CG	1000	30	bad
ecoli	EC	336	10	imU
haberman	HA	306	26	died
hepatitis	HE	155	21	die
ionosphere	IO	351	36	bad
new-thyroid	NT	215	16	hyper
pima	PI	768	35	positive
solar-flare	SF	1066	4	F
transfusion	TR	748	24	yes
vehicle	VE	846	23	van
yeast	YE	1484	3	ME2

of imbalance and that were used in other related works. In case of datasets with more than one majority class, we aggregated them into one class. Their characteristics are given in Table 1.

The experiments were run with a stratified 10-fold cross-validation repeated five times for better reproducibility of the results and to reduce a possible variance of estimating the average of the measures.

In the first part of experiments we compared single classifiers with all classification strategies used with rules sets induced by MODLEM. Table 2 presents G-mean and sensitivity for 5 classification strategies: Grzymala (denoted as GRZ), default rule (DEF), nearest rule (NR), discrimination measure (DM) and m-estimate (ME). One can notice that DM strategy was better than other strategies on both measures. To evaluate more precisely the differences between all these classification strategies, we applied a non-parametric ranked Friedman test, which globally compares the performance of k classifiers on m data sets. We analysed the values of sensitivity (see right-hand part of Table 2). A value of the statistics was very high (417.6) – much greater than the critical value 2.53 (for confidence level 0.05). So, we could reject the null-hypothesis saying that all compared single classifiers perform equally well. Then, we carried out a post-hoc analysis of differences between the average ranks of classifiers (the lower rank, the better classifier). These ranks are: 1.18 (DM), 3.15 (DEF), 3.09 (GRZ), 4.55 (ME) and 3.03 (NR). The critical difference CD (according to Nemenyi) is 1.48 – so we can say that DM performs significantly better than other classification strategies and m-estimate is the worst option.

Table 2. G-mean and Sensitivity for the single classifier

	G-mean					Sensitivity				
Data	**DM**	**DEF**	**GRZ**	**ME**	**NR**	**DM**	**DEF**	**GRZ**	**ME**	**NR**
AC	0.878	0.865	0.862	0.860	0.865	0.843	0.793	0.795	0.785	0.800
BC	0.556	0.519	0.489	0.473	0.507	0.452	0.325	0.294	0.264	0.314
BU	0.679	0.638	0.658	0.655	0.660	0.628	0.499	0.538	0.538	0.566
CA	0.943	0.888	0.893	0.909	0.893	0.894	0.789	0.799	0.826	0.797
CL	0.360	0.313	0.245	0.228	0.250	0.154	0.103	0.063	0.054	0.066
CM	0.600	0.465	0.466	0.436	0.461	0.478	0.243	0.244	0.211	0.239
CG	0.642	0.572	0.571	0.534	0.554	0.561	0.373	0.371	0.318	0.350
EC	0.759	0.645	0.639	0.634	0.634	0.623	0.429	0.420	0.414	0.411
HA	0.512	0.438	0.458	0.426	0.466	0.361	0.230	0.252	0.214	0.263
HE	0.690	0.595	0.604	0.540	0.594	0.598	0.375	0.394	0.313	0.378
IO	0.771	0.841	0.837	0.553	0.806	0.753	0.771	0.718	0.439	0.787
NT	0.919	0.900	0.905	0.913	0.906	0.860	0.823	0.826	0.843	0.831
PI	0.559	0.617	0.598	0.453	0.391	0.559	0.460	0.440	0.457	0.184
SF	0.396	0.267	0.280	0.230	0.235	0.172	0.072	0.079	0.053	0.056
TR	0.566	0.483	0.474	0.459	0.485	0.453	0.261	0.250	0.233	0.263
VE	0.950	0.915	0.917	0.910	0.914	0.953	0.854	0.858	0.845	0.854
YE	0.573	0.427	0.441	0.406	0.434	0.343	0.184	0.196	0.167	0.190

Then, we analysed a standard version of bagging. We observed that bagging improved the recognition of a majority class (i.e. specificity). However, systematic improvements were not observed for the minority class (sensitivity) – bagging sometimes improved the performance (e.g. for ionosphere), or deteriorated the results (e.g. solar-flare), which could influence the G-mean measure (Table 3). We performed a Wilcoxon signed rank test to examine the importance of differences on G-mean between bagging and a single classifier for each classification strategy. Null hypothesis (equal performance) could not be rejected – p-values on all strategies were about 0.8. This result suggests that using standard bagging is not sufficient for dealing with imbalanced datasets.

Table 4 presents the G-mean and sensitivity measures for bagging classifier with abstaining. One can notice that abstaining improves the results compared to both single and bagging classifiers (Tables 2 and 3). The difference is even more visible for the sensitivity measure. We again performed the Wilcoxon test for each classification strategy with respect to G-mean measure – see Table 5. For all strategies (except DM for very low confidence level) the differences between bagging with abstaining and other classifiers were statistically significant (p-value < 0.01). The same refered to Wilcoxon test on the sensitivity measure, which we do not present in details – the p-values were even smaller than for G-mean, but the null hypothesis for DM, comparing abstaining with a single classifier, could not be rejected. DM improves more the sensitivity, while it seems to be the worst strategy on specificity. Friedman test on sensitivity for bagging with abstaining showed that DM strategy had the best average rank, while the differences between other strategies were statistically insignificant.

Table 3. G-mean and Sensitivity for the bagging classifier

	G-mean					Sensitivity				
Data	**DM**	**DEF**	**GRZ**	**ME**	**NR**	**DM**	**DEF**	**GRZ**	**ME**	**NR**
AC	0.860	0.886	0.884	0.887	0.881	0.810	0.823	0.820	0.823	0.820
BC	0.537	0.499	0.483	0.476	0.492	0.404	0.281	0.266	0.249	0.274
BU	0.687	0.683	0.697	0.690	0.698	0.602	0.515	0.547	0.547	0.561
CA	0.933	0.910	0.913	0.915	0.914	0.876	0.829	0.835	0.838	0.836
CL	0.355	0.168	0.106	0.004	0.168	0.143	0.029	0.011	0.000	0.029
CM	0.607	0.425	0.433	0.382	0.430	0.462	0.194	0.201	0.155	0.199
CG	0.651	0.550	0.555	0.501	0.561	0.547	0.321	0.326	0.260	0.333
EC	0.758	0.640	0.635	0.604	0.654	0.610	0.417	0.411	0.371	0.437
HA	0.520	0.425	0.415	0.360	0.412	0.356	0.212	0.204	0.147	0.198
HE	0.733	0.584	0.638	0.584	0.607	0.673	0.359	0.431	0.359	0.394
IO	0.913	0.906	0.904	0.906	0.905	0.852	0.834	0.826	0.831	0.829
NT	0.922	0.917	0.920	0.919	0.919	0.861	0.849	0.854	0.849	0.849
PI	0.678	0.662	0.655	0.662	0.662	0.561	0.483	0.472	0.485	0.484
SF	0.367	0.198	0.173	0.173	0.159	0.145	0.040	0.030	0.030	0.026
TR	0.566	0.484	0.475	0.433	0.470	0.443	0.256	0.247	0.202	0.241
VE	0.945	0.936	0.935	0.932	0.937	0.938	0.890	0.889	0.883	0.893
YE	0.564	0.400	0.388	0.381	0.380	0.325	0.161	0.151	0.145	0.145

Table 4. G-mean and Sensitivity for the bagging classifier with abstaining

	G-mean					Sensitivity				
Data	**DM**	**DEF**	**GRZ**	**ME**	**NR**	**DM**	**DEF**	**GRZ**	**ME**	**NR**
AC	0.881	0.886	0.894	0.894	0.895	0.833	0.833	0.848	0.848	0.848
BC	0.577	0.537	0.546	0.555	0.545	0.441	0.345	0.356	0.356	0.356
BU	0.709	0.704	0.704	0.715	0.715	0.638	0.603	0.601	0.601	0.616
CA	0.973	0.974	0.979	0.979	0.976	0.954	0.949	0.959	0.959	0.954
CL	0.372	0.167	0.190	0.191	0.211	0.149	0.029	0.037	0.037	0.046
CM	0.566	0.503	0.495	0.500	0.500	0.379	0.280	0.270	0.270	0.276
CG	0.684	0.622	0.626	0.639	0.624	0.557	0.427	0.432	0.432	0.430
EC	0.772	0.722	0.711	0.711	0.721	0.620	0.537	0.520	0.520	0.534
HA	0.534	0.490	0.478	0.488	0.486	0.375	0.296	0.281	0.281	0.290
HE	0.724	0.709	0.717	0.718	0.717	0.600	0.563	0.572	0.572	0.571
IO	0.918	0.912	0.911	0.909	0.907	0.866	0.848	0.848	0.844	0.841
NT	0.928	0.917	0.928	0.930	0.932	0.871	0.846	0.869	0.869	0.874
PI	0.721	0.711	0.707	0.685	0.707	0.642	0.593	0.588	0.534	0.584
SF	0.423	0.275	0.287	0.287	0.283	0.184	0.077	0.084	0.084	0.081
TR	0.516	0.486	0.487	0.490	0.495	0.358	0.308	0.306	0.306	0.319
VE	0.964	0.952	0.953	0.956	0.953	0.959	0.929	0.931	0.931	0.929
YE	0.590	0.446	0.437	0.437	0.442	0.353	0.200	0.192	0.192	0.196

Table 5. p-value in paired Wilcoxon test - Abstaining vs other classifiers on G-mean

Abstaining vs	**DM**	**DEF**	**GRZ**	**ME**	**NR**
Single	0.0216	0.00486	0.00193	0.00071	0.0012
Bagging	0.0035	0.0006	0.000294	0.000284	0.000294

Another observation was that for more "majority-biased" strategies – DEF, GRZ, ME and NR – abstaining gives quite comparable results for a given dataset, even if these strategies behave differently in single classifiers – see, e.g., ionosphere, where ME gives the worst result in a single classifier, but abstaining improves it so that it becomes comparable to other strategies (compare right-hand parts of Table 2 and 4). The same observation may be drawn for pima dataset and NR strategy, or for bupa and DEF strategy.

6 Conclusions

In this paper we discussed the idea of abstaining in component classifiers by excluding partial matching for unordered sets of rules and its impact on learning from imbalanced data.

Let us summarize and discuss the results of the experiments. First of all, we conclude that introducing abstaining of classifiers in bagging has improved the recognition of minority class. The extensive comparative study on 17 datasets has showed that our proposal is significantly better than a single classifier and classical bagging (without abstaining), for both sensitivity and G-mean measures. An improvement on these two measures is more visible than on the global accuracy, which was studied in the previous paper [2]. Standard bagging is not better than a single classifier when sensitivity and G-mean are considered, but it improves the specificity measure (referring to better recognizing the majority classes).

We conducted the experiments using 5 classification strategies. The behaviour of four strategies – DEF, GRZ, ME and NR – was different to DM. These strategies are more prone to give incorrect decisions by assigning examples to majority classes too often. Their solving of no-matching seems to be biased too much toward majority classes. In case of bagging these incorrect predictions may outweight the correct ones when establishing the final outcome of the ensemble. Refraining from making a prediction by component classifiers when no rule has been matched may solve such situations, which was demonstrated by our experiments where abstaining clearly improved performance of the ensemble, confirming our intuition.

DM strategy, on the other hand, seems to be less biased toward majority classes (see its description in 3.2 – rules are evaluated by another measure than supports, so weaker minority rules may avoid being dominated by more general and stronger majority rules). It recognizes the minority class significantly better than the remaining strategies, which we confirmed in a Friedman test on sensitivity for single classifier and bagging with abstaining. This may also explain why DM's result was not so improved by introducing bagging with abstaining.

In the future work we would like to confirm these observations for abstaining by testing another rule learning algorithm and to modify a multiple matching part of a classification strategy to produce unknown answer in case of uncertainty between two competitive class assignments.

Acknowledgments. The research has been supported by the Ministry of Science and Higher Education, grant no. N N519 441939.

References

1. An, A.: Learning classification rules from data. Computers and Mathematics with Applications 45, 737–748 (2003)
2. Blaszczynski, J., Stefanowski, J., Zajac, M.: Ensembles of Abstaining Classifiers Based on Rule Sets. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS (LNAI), vol. 5722, pp. 382–391. Springer, Heidelberg (2009)
3. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
4. Clark, P., Boswell, R.: Rule Induction with CN2: Some Recent Improvements. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 151–163. Springer, Heidelberg (1991)
5. Cestnik, B.: Estimating probabilities: A crucial task in Machine Learning. In: Proc. of the 9th European Conf. on Artificial Intelligence (ECAI 1990), pp. 147–150 (1990)
6. Cohen, W., Singer, Y.: A simple, fast and effective rule learner. In: Proc. of the 16th National Conference on Artificial Intelligence AAAI 1999, pp. 335–342 (1999)
7. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 99, 1–22 (2011)
8. Grzymala-Busse, J.W.: Managing uncertainty in machine learning from examples. In: Proc. 3rd Int. Symp. in Intelligent Systems, pp. 70–84 (1994)
9. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Data and Knowledge Engineering 21(9), 1263–1284 (2009)
10. Rucket, U., Kramer, S.: Towards tight bounds for rule learning. In: Proc. of the 21st Int. Conf. on Machine Learning, ICML 2004, pp. 711–718 (2004)
11. Stefanowski, J.: On Combined Classifiers, Rule Induction and Rough Sets. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 329–350. Springer, Heidelberg (2007)
12. Stefanowski, J., Wilk, S.: Improving Rule Based Classifiers Induced by MODLEM by Selective Pre-processing of Imbalanced Data. In: Proc. of the RSKD Workshop at ECML/PKDD, Warsaw, pp. 54–65 (2007)
13. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. J. Artificial Intelligence Research 6, 1–34 (1997)

Semi-supervised Ensemble Learning of Data Streams in the Presence of Concept Drift

Zahra Ahmadi and Hamid Beigy

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
{z_ahmadi,beigy}@ce.sharif.com

Abstract. Increasing access to very large and non-stationary datasets in many real problems has made the classical data mining algorithms impractical and made it necessary to design new online classification algorithms. Online learning of data streams has some important features, such as sequential access to the data, limitation on time and space complexity and the occurrence of concept drift. The infinite nature of data streams makes it hard to label all observed instances. It seems that using the semi-supervised approaches have much more compatibility with the problem. So in this paper we present a new semi-supervised ensemble learning algorithm for data streams. This algorithm uses the majority vote of learners for the labeling of unlabeled instances. The empirical study demonstrates that the proposed algorithm is comparable with the state-of-the-art semi-supervised online algorithms.

Keywords: Stream Mining, Concept Drift, Ensemble Learning, Semi-Supervised Learning.

1 Introduction

The growing availability of data in web has made mining and knowledge discovery from huge amounts of data, difficult and of interest. As the amount of data is very large (and ideally infinite) it cannot be stored and therefore there is a need for new algorithms to process the stream of data online. This is called stream mining and it has been a challenging problem in recent years. Data streams have some important properties [1]:

- There should be a forgetting mechanism as the received data could not be stored completely. The most common way of forgetting is to use a window of constant size. However, adaptive window [2] or density based forgetting [3, 4] is also presented.
- Time and algorithmic complexity should be simple as the data must be processed online.
- The most important property of data streams is concept drift, which is the change in the feature or class distribution over the time:

$$P(X, C) = P(C|X) \times P(X) \tag{1}$$

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 526–537, 2012.

where X is the feature vector and C is the class label. If the drift occurs in feature space (P(X)), it is called virtual drift but if it occurs in target function (P(C|X)), it is called real drift. We only consider the occurrence of the drift, change of joint probability of P(X,C) over time, no matter whether the drift is virtual or real.

The concept drift could be abrupt, gradual or recurring [5]. Whether the drift is the underlying distribution of data changes suddenly at time t, abrupt drift has occurred. If the distribution changes in a period of time (not at a specific time), and the probability of new distribution increases gradually, the drift is called gradual. If the previously seen concepts reappear some time later, they are called recurring concepts. As there should be a forgetting mechanism in data streams to handle the drifts, the previously seen concepts may be forgotten and so the true classification of recurring concepts is an important ability of data stream algorithms.

There have been extensive studies on the supervised learning of data streams in the presence of abrupt, gradual or recurring concepts. However, the semi-supervised approaches are not considered much and just a few researches are done recently [6-13], so the problem is identified challenging.

This paper proposes an ensemble learning method to classify the instances and predict the labels of unlabeled instances. For each classifier in the ensemble, the majority vote of other classifiers is used to label the unlabeled instances and then it is used to update the classifier. It is proven that even the labeling process is noisy, the classification is PAC learnable. The results show the effectiveness of our algorithm in terms of accuracy in comparison to one of the promising ensemble algorithms in the literature of semi-supervised data streams.

The structure of the paper is as follows: in the next section the related works of semi-supervised data streams are discussed. In section 3 the proposed algorithm is presented and in section 4 the experimental results and evaluation of the method is presented. Section 5 concludes the paper and discusses some of the future works.

2 Related Works

As data streams are infinite, arrive continuously and there should be online classification, labeling all of the arrived data is impossible. So in the recent years, there is a motivation on semi-supervised learning of data streams. Few algorithms have been presented to classify scarcely labeled data streams [6-13]. The algorithms could be categorized in two groups according to the number of classifiers used in the learning process: single [6, 8-10] and ensemble [12-14]. The semi-supervised approaches are categorized in one of the following methods:

- Using K-means clustering algorithm to label the unlabeled instances [6, 10, 12]. K-means is used because of its simplicity and efficiency.
- Using expectation maximization algorithm to estimate the label of instances [9].

The first semi-supervised learning algorithm of data streams was presented by Klinkenberg [8] and used the SVM and window adjustment approach. Later, another algorithm based on relational k-means transfer semi-supervised support vector machines (RK-TS^{3}VM) was proposed in [10].

The algorithm presented in [6], extends online decision trees to support recurring concepts. It uses k-means to cluster and label the instances. To cover the recurring concepts, it uses the conceptual clusters in the leaves of the trees. To avoid overfitting, pruning is done regularly.

Another algorithm which uses ensemble learning is presented in [12]. In each window (or batch of data), the constraint-based clustering algorithm is applied and K homogeneous clusters are created. A homogeneous cluster is a cluster which contains only unlabeled instances or only labeled instances of a single class. Some information about each micro-cluster (centroid, number of instances,…) is maintained as pseudo-points. Then label propagation is done on the pseudo-points and these points act as a classification model. The ensemble is kept up to date with the current concept and periodically refines the L classifiers to cope with the drift. The refinement is done according to the accuracy of the learners on the current batch. We compared our method to this algorithm because of their similar approach.

On the other hand, there are several other approaches in the literature of semi-supervised learning: self-training, probabilistic generative models, cluster then labeling, co-training, graph based approaches and transductive support vector machines.

Our proposed algorithm could be categorized as a self-training model, but it has some differences in the regular methods of self-training. In regular self-training approaches, the learner first learns from labeled data and then uses its prediction on unlabeled instances and selects some instances with the more confident labels. The new labeled instances are added to the training set and the learning process is repeated again. This process repeats iteratively until no new instance is added to the training set. However in our proposed algorithm, the algorithm should act online and so the iterative process is omitted and the labeling process is done once. The advantage of this approach is its simplicity and the fact that it is a wrapper method, so it could be applied on different learning algorithms.

3 The SSEL Algorithm

As we discussed previously, labeling of all the instances in the stream is impossible. So the approach we follow here is a semi-supervised approach, which assumes that among arrived instances, some of them are labeled randomly. An ensemble method is used to label the unlabeled instances to improve the performance of classification. The proposed algorithm is presented in Table 1.

When a new window of data arrives, first of all, the labeled instances are separated from the unlabeled ones and the ensemble learners are updated with the labeled instances. Assume that the number of learners is K. Diversity is an important feature of ensemble learning and in the learning of data streams it plays an even more

important role. If we use all the instances in the window, after some time the base learners become the same, so we use bootstrapping to have diverse learners. Then the process of using unlabeled data begins. For each learner, we determine a set of labeled instances from unlabeled data. To do this, we use the majority vote of other learners for the specified instance. We use the predicted label of K-1 learners (all learners except the one which we are selecting the instances for), if their ensemble does better than a random classifier (which is checked in the line 17 of pseudo code). If the prediction of K-1 learners is correct, then the K^{th} learner has received a valid labeled instance. Otherwise the label will be noisy. In the worst case, when we have noisy instances, if the number of labeled instances is sufficient, we could decrease the error rate of classification. To do this, we used the idea from [15].

Assume the number of misclassified instances in the t^{th} window is $\eta_L|L_t|$, where L_t is the set of labeled instances in the t^{th} window and η_L is the noise rate of the classification algorithm. e_i^t is the noise rate upper bound of h_k ensemble of classifiers for all base classifiers with $k \neq i$ in the window t. If z is the number of unlabeled instances having the vote of more than 50% of the learners' and z' is the number of correct classified instances; then we could estimate e_i^t by $\frac{z-z'}{z}$. Here z could be written as $|W^t|-|L_t|-|f^t|$, where W^t is the t^{th} window and f^t is the set of unlabeled instances that the ensemble of K-1 base learners has a 50%-50% vote on them. So the number of misclassified instances from unlabeled data will be $e_i^t|L^t|$, where L^t is the set of instances from unlabeled data which are labeled in the window t ($L^t = W^t-L_t-f^t$). Therefore we could write the noise rate in the t^{th} window as:

$$\eta^t = \frac{\eta_L|L_t| + e_i^t|L^t|}{|L_t \cup L^t|} \tag{2}$$

From [16], if a sequence σ of m samples is drawn, where the minimum number of instances should be computed by

$$m = \frac{2\ln\left(\frac{2N}{\delta}\right)}{(\varepsilon)^2(1-2\eta)^2} \tag{3}$$

where N is the number of hypotheses, δ is the appropriate confidence, ε is the worst case of classification error rate and η is an upper bound of classification error rate. Then a hypothesis (H_i) minimizes disagreement on σ will be PAC learnable (H^* is a ground truth hypothesis):

$$P\left(distance\left(H_i, H^*\right) \geq \varepsilon\right) \leq \delta \tag{4}$$

Using equation (3) in the process of data stream learning, the minimum number of instances in each window should at least be:

$$m^t = \frac{2\ln\left(\frac{2N}{\delta}\right)}{(\varepsilon^t)^2(1-2\eta^t)^2} \tag{5}$$

Table 1. Semi-Supervised Ensemble Learning Algorithm (SSEL)

Input: $\theta = \frac{1}{C}$, where C is the number of classes,
K : Number of weak classifiers,
data stream in the window size of w,
LearnIncremental: Learning algorithm.

for j=1..K **do**
  $e'_j = 0.5$ //learner's error
  $l'_j = 0$ //length of unlabeled set in the window
end for
while true **do**
  receive w_i window of data
  L_i = separate labeled data of w_i
  U_i = separate unlabeled data of w_i
  Test current hypothesis on L_i
  for j=1..K **do**
    S_{ji} = BootstrapSample(L_i)
    h_{ji} = LearnIncremental($h_{j(i-1)}$, S_{ji})
  end for
  for j=1..K **do**
    $L_{ji} = \emptyset$ //the set of unlabeled instances and their predicted labels
    e_{ji} = MeasureError($h_{ki}|_{k=1}^{K}$) (k≠i) //estimate the classification error rate
    if ($e_{ji} < \theta$) **then** //if the ensemble works better than random
      for every x∈U_i **do**
        if most of h_{ki}(x) (k≠i) classify x in c **then** $L_{ji} = L_{ji} \cup \{(x,c)\}$
      end for
      if ($l'_j = 0$) **then** $l'_j = \left\lfloor \frac{e_{ji}}{e'_j - e_{ji}} + 1 \right\rfloor$ //in the first window
      if($\frac{e_{ji}}{e'_j - e_{ji}} < l'_j < |L_{ji}|$) **then** //the condition of equation 8-9
//subsampling is done if number of unlabeled set is more than the maximum size
        L_{ji} = Subsample(L_{ji}, $\left\lceil \frac{e'_j l'_j}{e_{ji}} - 1 \right\rceil$)
  end for
  for j=1..K **do**
    if ($L_{ji} \neq \emptyset$) **then**
      h_{ji} = LearnIncremental(L_{ji})
      $e'_j = e_{ji}$
      $l'_j = |L_{ji}|$
  end for
end while

As the goal of an online learner is to become better in time, ε^t should decrease as t survives ($\varepsilon^t < \varepsilon^{t-1}$). Using the aforementioned equation we have:

$$\mathrm{m}^{\mathrm{t}}(1-2\eta^t)^2 > \mathrm{m}^{\mathrm{t}-1}(1-2\eta^{t-1})^2 \tag{6}$$

Having equation (2) in hand and $\mathrm{m}^{\mathrm{t}} = |\mathrm{L_t} \cup \mathrm{L}^{\mathrm{t}}|$, we can substitute (6):

$$|\mathrm{L_t} \cup \mathrm{L}^{\mathrm{t}}|(1-2\frac{\eta_{\mathrm{L}}|L_t| + e_i^t|L^t|}{|\mathrm{L_t} \cup \mathrm{L}^{\mathrm{t}}|})^2 > |\mathrm{L_{t-1}} \cup \mathrm{L}^{\mathrm{t}-1}|(1-2\frac{\eta_{\mathrm{L}}|L_{t-1}| + e_i^{t-1}|L^{t-1}|}{|\mathrm{L_{t-1}} \cup \mathrm{L}^{\mathrm{t}-1}|})^2$$

$$\frac{(|\mathrm{L_t} \cup \mathrm{L}^{\mathrm{t}}| - 2\eta_{\mathrm{L}}|L_t| - 2e_i^t|L^t|)^2}{|\mathrm{L_t} \cup \mathrm{L}^{\mathrm{t}}|} > \frac{(|\mathrm{L_{t-1}} \cup \mathrm{L}^{\mathrm{t}-1}| - 2\eta_{\mathrm{L}}|L_{t-1}| - 2e_i^{t-1}|L^{t-1}|)^2}{|\mathrm{L_{t-1}} \cup \mathrm{L}^{\mathrm{t}-1}|} \tag{7}$$

By making some simplification assumptions, such as fixed length of window over the time ($\mathrm{W}^{\mathrm{t}} = \mathrm{W}^{\mathrm{t}-1}$) and fixed number of labeled instances in each window ($|L_{t-1}| = |L_t|$), two conditions should be satisfied:

$$\begin{cases} |\mathrm{L_t} \cup \mathrm{L}^{\mathrm{t}}| < |\mathrm{L_{t-1}} \cup \mathrm{L}^{\mathrm{t}-1}| \\ |\mathrm{L_t} \cup \mathrm{L}^{\mathrm{t}}| - 2\eta_{\mathrm{L}}|L_t| - 2e_i^t|L^t| > |\mathrm{L_{t-1}} \cup \mathrm{L}^{\mathrm{t}-1}| - 2\eta_{\mathrm{L}}|L_{t-1}| - 2e_i^{t-1}|L^{t-1}| \end{cases}$$

$$\begin{cases} |\mathrm{f}^{\mathrm{t}}| > |\mathrm{f}^{\mathrm{t}-1}| \\ |\mathrm{W}| - |\mathrm{f}^{\mathrm{t}}| - 2\eta_{\mathrm{L}}|L_t| - 2e_i^t|L^t| > |\mathrm{W}| - |\mathrm{f}^{\mathrm{t}-1}| - 2\eta_{\mathrm{L}}|L_t| - 2e_i^{t-1}|L^{t-1}| \\ 2e_i^{t-1}|L^{t-1}| - 2e_i^t|L^t| > |\mathrm{f}^{\mathrm{t}}| - |\mathrm{f}^{\mathrm{t}-1}| > 0 \end{cases} \Longrightarrow$$

So the number of unlabeled instances that are labeled by the algorithm should satisfy the following inequality:

$$0 < \frac{e_{\mathrm{i}}^{\mathrm{t}}}{e_{\mathrm{i}}^{\mathrm{t}-1}} < \frac{|L^{t-1}|}{|L^t|} < 1 \tag{8}$$

By substituting $e_{\mathrm{i}}^{\mathrm{t}} = \frac{|\mathrm{L}^{\mathrm{t}}| - \mathrm{z}'^{\mathrm{t}}}{|\mathrm{L}^{\mathrm{t}}|}$ in (8), we conclude:

$$0 < \frac{|\mathrm{L}^{\mathrm{t}}| - \mathrm{z}'^{\mathrm{t}}}{|\mathrm{L}^{\mathrm{t}-1}| - \mathrm{z}'^{\mathrm{t}-1}} < 1 \tag{9}$$

We could find the maximum number of unlabeled instances in the t^{th} window to be labeled, from (8) and (9):

$$|L^t| = \left\lceil \frac{e_{\mathrm{i}}^{\mathrm{t}-1}|L^{t-1}|}{e_{\mathrm{i}}^{\mathrm{t}}} - 1 \right\rceil = |\mathrm{L}^{\mathrm{t}-1}| + \mathrm{z}'^{\mathrm{t}} - \mathrm{z}'^{\mathrm{t}-1} \tag{10}$$

So if the number of unlabeled instances to be labeled in the t^{th} window becomes more than the value obtained in (10), the subsampling process is done (line 22 and 23 in Table 1). Finally a set of instances is determined for each of base learners to be updated with. This process is repeatedly done on data windows.

4 Experimental Results

To evaluate the proposed algorithm, we first introduce some datasets which contain different kinds of concept drifts. Then the proposed algorithm is compared with one of the most promising semi-supervised data stream algorithms called ReaSC [12]. The experiments show the effectiveness of the proposed algorithm.

4.1 Data Sets

We have used six datasets where two of them are artificial and the others are real. We tried to select datasets containing all types of concept drift: abrupt, gradual and recurring concepts.

The SEA [17] dataset is an artificial dataset which consists of abrupt drift and has 2 classes and 3 attributes of values between 0 and 10. The dataset we used here consists of 50,000 instances.

Another artificial dataset is Hyperplane [18], which has 100,000 instances with gradual drift. The concept is defined by

$$f(x) = \sum_{j=1}^{d-1} a_j . \frac{(x_j + x_{j+1})}{x_j} \tag{11}$$

where the value of a_j controls the shape of the decision surface and f(x) is the class label of instance x. Concept drift is controlled through the following parameters: (1) t, controls the magnitude of the concept drift; (2) p, controls the number of attributes whose weights are involved in the drift; and (3) h and $g \in \{-1, 1\}$, control the weight adjustment direction for attributes involved in the change. For each instance x, a_i is adjusted by $g*t / M$ where after receiving M instances, there is an h percentage of chance that weight change will inverse its direction, *i.e.*, $g = -g$. Here the Hyperplane dataset has five classes and 10 attributes. p is set to 5, and t to 0.1 in every M=2000 instances and weight adjustment inverses the direction with h=20% of chance.

Another dataset used in the experiments is the famous KDD Cup 99 [19], which has 23 class labels and 42 attributes. Here we used 10% of the whole dataset so it has 492,000 instances. One important feature of this dataset is the occurrence of novelty which means that new classes appear through the time.

Usenet dataset used in [20] contains a stream of emails with different topics, where the user labels them as interesting or junk. The data in Usenet posts [19] has been used to construct this dataset. Three topics from 20 newsgroups are selected. The user is interested in one or two topics in each concept and so he/she labels the emails according to his/her interest. User interests can change in time, so this dataset contains recurring concepts and abrupt concept drift (Table 2). The dataset consists of 1500 instances and 913 attributes and is divided into 5 time periods each having equal number of instances.

Sensor dataset [17] is a real dataset which consists of the information collected from 54 sensors deployed in Intel Berkeley Research laboratory in a two-month period. The class label is the sensor ID, so there are 54 classes, 5 attributes and 2,219,803 instances. There are some kinds of concept drifts in this dataset. For example, lighting (or the temperature of some specific sensors) during the working hours is much stronger than nights.

Table 2. Usenet Dataset concepts

	1-300	300-600	600-900	900-1200	1200-1500
Medicine	+	-	+	-	+
Space	-	+	-	+	-
Baseball	-	+	-	+	-

Elec [21] is another dataset gathered from real data. This dataset consists of the price and the demand on an electricity store which is gathered every 30 minutes. The dataset contains 45,312 instances and 8 attributes and the label shows the change in the price in comparison to the mean price of previous 24 hours.

4.2 Implementation and Parameter Setting

The proposed algorithm was developed by Java using WEKA [22] and MOA [23] environments. As discussed previously, there is a need for incremental classifier. Here we used the incremental decision tree as the base classifier of ensemble learners. Decision trees are appropriate classifiers for online learning as they are fast and accurate learners. The number of base classifiers is set to three.

In ReaSC algorithm, another parameter is the number of possible clusters. Our proposed method does not use clustering and thus has no need of such a parameter. This is an advantage since clustering is time consuming and determining the cluster parameter is not easy, however, it has an impact on the classification accuracy. This parameter is set to 50 according to the authors' experiments in ReaSC [12].

In all datasets, we assume that 20% of instances in each window are labeled randomly and the window size is set to 1000 instances, except for the Usenet dataset in which the window size is set to 100 ones (as the number of instances in it is small).

4.3 Results and Discussion

We compared our method with the ReaSC algorithm [12] in terms of cumulative accuracy through the windows of data. We used cumulative accuracy to obtain smoother plots. The results of our experiments on the datasets are shown in figures 1 to 6. The results on the datasets containing different types of concept drift are definitely promising. As it can be seen in SEA dataset, there is a difference of at least 3% in performance. In Hyperplane dataset the difference is much more and about 15% of improvement in accuracy. So it seems that our proposed algorithm works much better in datasets with gradual drifts. It can be due to the boosting nature of the algorithm.

In real datasets, the type and place of concept drift is not determined. But in all datasets (KDD CUP99, Usenet, Sensor and Elec) our proposed algorithm has better performance. Especially in Sensor dataset we have a performance improvement of about 45% and in Elec, it is about 15%. These results are approximately remained the same even with different values of parameters (number of base learners, percentage of labeled data and window size).

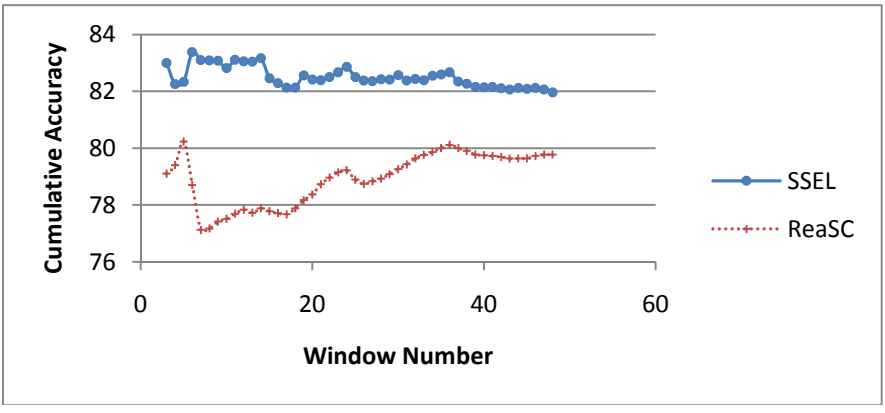


Fig. 1. Total accuracy of SSEL and ReaSC algorithms in SEA dataset

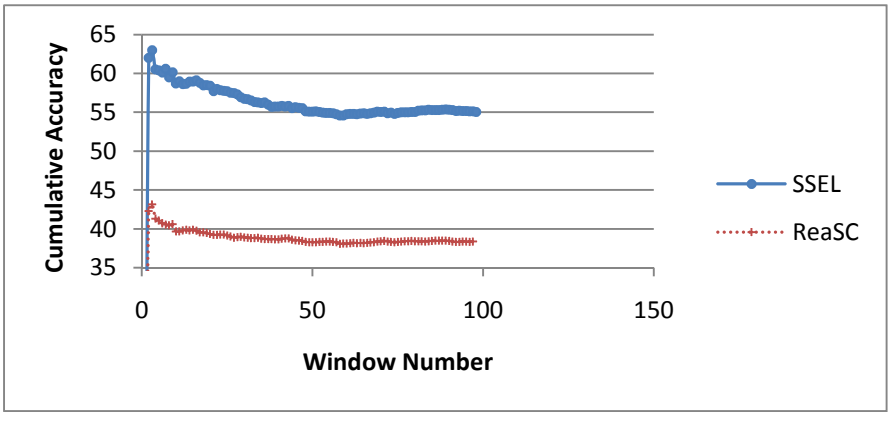


Fig. 2. Total accuracy of SSEL and ReaSC algorithms in HyperPlane dataset

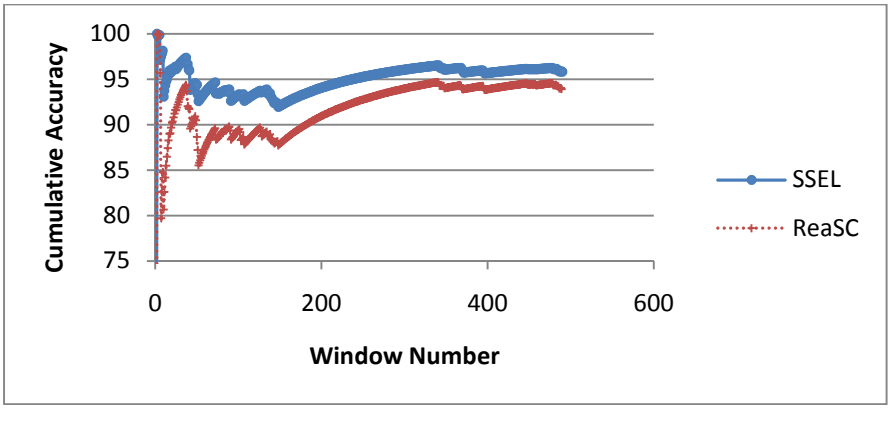


Fig. 3. Total accuracy of SSEL and ReaSC algorithms in KDD CUP99 dataset

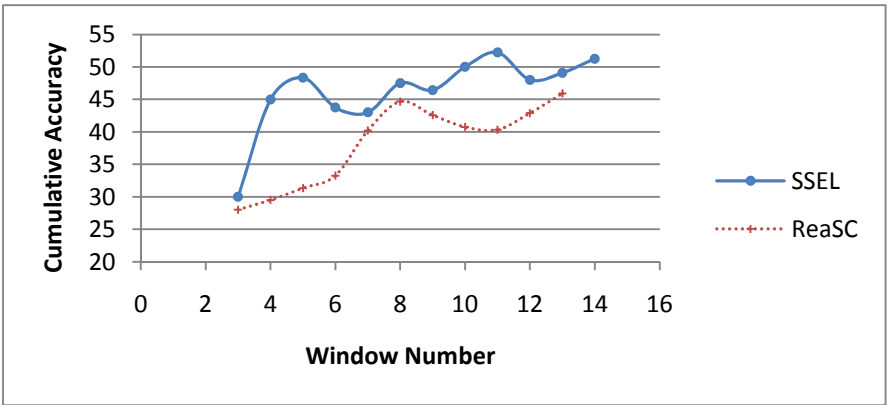


Fig. 4. Total accuracy of SSEL and ReaSC algorithms in Usenet dataset

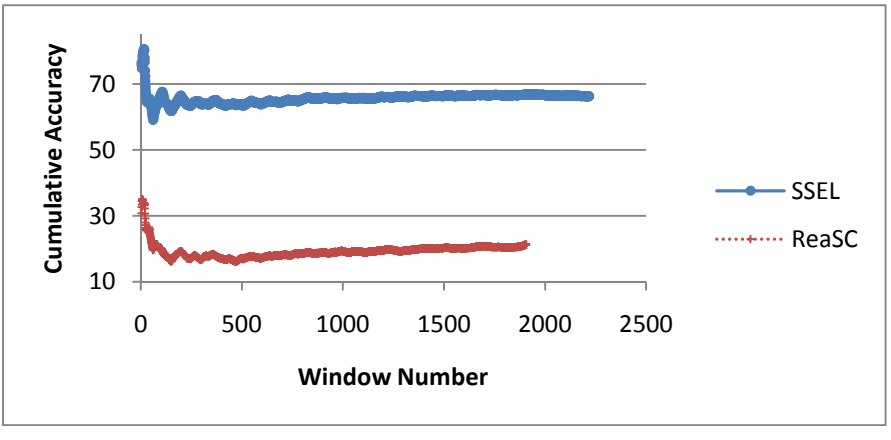


Fig. 5. Total accuracy of SSEL and ReaSC algorithms in the Sensor dataset

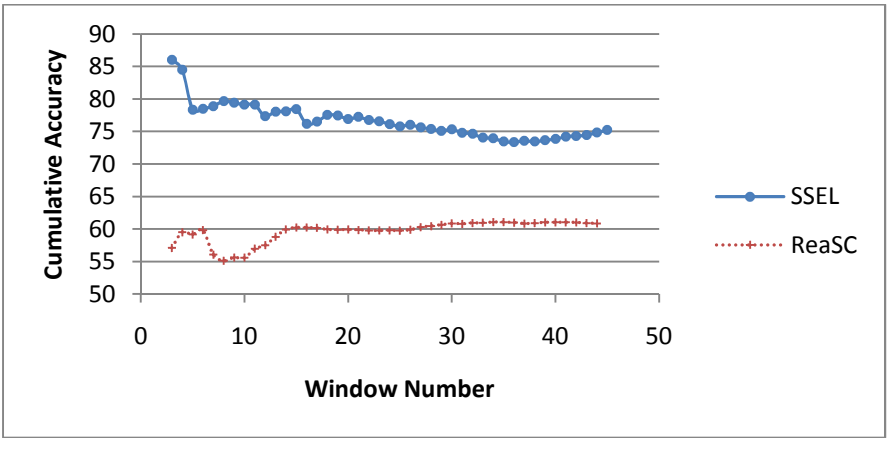


Fig. 6. Total accuracy of SSEL and ReaSC algorithms in the Elec dataset

All in one, we can see that our proposed algorithm works much better than the other algorithm having a similar approach in semi-supervised ensemble learning.

5 Conclusion and Future Works

In this paper we have proposed a new semi-supervised ensemble learning (SSEL) algorithm for the classification of streaming data. The approach could be categorized as modified self-training but without the conventional problems of self-training. In self-training methods, if the learners are weak and predict the label of unlabeled instances incorrectly, using the unlabeled data will degrade the performance of the learner. But here in SSEL we showed that if the number of instances in each window is enough (according to the equation (10)), then the algorithm is PAC learnable and noise will not degrade the performance of the learner.

For future works we could develop the experiments and examine more datasets and different parameters (the number of base classifiers, the percentage of labeled data and the window size) to get more reliable results. On the other hand, if the drift rate becomes fast, our method has delay in detecting the changes and the performance decrease. Also the upper bound of drift rate could be calculated as a future work.

References

1. Tsymbal, A.: The Problem of Concept Drift: Definitions and Related Work (2004)
2. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23(1), 69–101 (1996)
3. Aha, D.W., Kibler, D., Albert, M.K.: Instance-Based Learning Algorithms. Mach. Learn. 6(1), 37–66 (1991)
4. Salganicoff, M.: Density-Adaptive Learning and Forgetting. In: Tenth International Conference on Machine Learning. Morgan Kaufmann (1993)
5. Zliobaite, I.: Learning under Concept Drift: an Overview (2010)
6. Li, P., Wu, X., Hu, X.: Mining Recurring Concept Drifts with Limited Labeled Streaming Data. In: 2nd Asian Conference on Machine Learning (ACML 2010). JMLR, Tokyo (2010)
7. Masud, M.M.: Adaptive Classification of Scarcely Labeled and Evolving Data Streams, in Computer Science, p. 161. The University of Texas, Dallas (2009)
8. Klinkenberg, R.: Using Labeled and Unlabeled Data to Learn Drifting Concepts. In: IJCAI 2001 Workshop on Learning from Temporal and Spatial Data. AAAI Press, Menlo Park (2001)
9. Borchani, H., Larrañaga, P., Bielza, C.: Mining Concept-Drifting Data Streams Containing Labeled and Unlabeled Instances. In: García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J.M., Ali, M. (eds.) IEA/AIE 2010, Part I. LNCS, vol. 6096, pp. 531–540. Springer, Heidelberg (2010)
10. Zhang, P., Zhu, X., Guo, L.: Mining Data Streams with Labeled and Unlabeled Training Examples. In: Proceedings of the 2009 Ninth IEEE International Conference on Data Mining. IEEE Computer Society (2009)

11. Widyantoro, D.H., Yen, J.: Relevant data expansion for learning concept drift from sparsely labeled data. IEEE Transactions on Knowledge and Data Engineering 17(3), 401–412 (2005)
12. Woolam, C., Masud, M.M., Khan, L.: Lacking Labels in the Stream: Classifying Evolving Stream Data with Few Labels. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS, vol. 5722, pp. 552–562. Springer, Heidelberg (2009)
13. Ditzler, G., Polikar, R.: Semi-supervised learning in nonstationary environments. IEEE
14. Kantardzic, M., Ryu, J.W., Walgampaya, C.: Building a New Classifier in an Ensemble Using Streaming Unlabeled Data. In: García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J.M., Ali, M. (eds.) IEA/AIE 2010, Part I. LNCS, vol. 6097, pp. 77–86. Springer, Heidelberg (2010)
15. Zhou, Z.-H., Li, M.: Tri-Training: Exploiting Unlabeled Data Using Three Classifiers. IEEE Trans. on Knowl. and Data Eng. 17(11), 1529–1541 (2005)
16. Angluin, D., Laird, P.: Learning From Noisy Examples. Machine Learning 2(4), 343–370 (1988)
17. 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. ACM, San Francisco (2001)
18. Zhu, X.: Stream Data Mining repository (2010), http://www.cse.fau.edu/~xqzhu/stream.html
19. Frank, A., Asuncion, A.: UCI Machine Learning Repository (2010), http://archive.ics.uci.edu/ml (cited May 2011)
20. Katakis, I., Tsoumakas, G., Vlahavas, I.: Tracking recurring contexts using ensemble classifiers: an application to email filtering. Knowledge and Information Systems 22(3), 371–391 (2009)
21. Harries, M.B., Sammut, C., Horn, K.: Extracting hidden context. Machine Learning 32(2), 101–126 (1998)
22. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2005)
23. Bifet, A., et al.: Moa: Massive online analysis. The Journal of Machine Learning Research 11, 1601–1604

Continuous User Feedback Learning for Data Capture from Business Documents

Marcel Hanke[1], Klemens Muthmann[1], Daniel Schuster[1], Alexander Schill[1], Kamil Aliyev[2], and Michael Berger[2]

[1] Computer Networks, Dept. of Computer Science, TU Dresden, Dresden, Germany
marcel.hanke@mailbox.tu-dresden.de
{klemens.muthmann,daniel.schuster,alexander.schill}@tu-dresden.de
[2] DocuWare AG, Germering, Germany
{kamil.aliyev,michael.berger}@docuware.com

Abstract. Automatically processing production documents requires document type detection as well as data capture to find appropriate index data from a post-OCR representation of the document. While current learning-based methods perform quite well due to many similar documents created with the same template, their machine learning models require intense training and are hard to update frequently. We provide a method for continuously incorporating user feedback in a layout-based extraction process taking care of both immediate learning as well as limiting the size of the model. The method is evaluated on a tagged corpus of more than 5,000 business documents. It allows not only continuous re-training of the model thus adapting it to new document templates, but also starting from scratch with an empty model requiring less than 10% of the corpus as training documents to reach an accuracy measure of more than 80%.

Keywords: User Feedback, Information Extraction, Document Classification, Document Archiving, Document Management.

1 Introduction

Document type classification, data capture, and detection of document sets are the three classical tasks in production document processing [8]. Existing solutions for doctype classification and data capture already achieve quite good extraction rates to find the right document type (e.g., invoice) as well as typical index data such as sender and receiver, document date, document id, or total amount. But as they are based on machine learning algorithms like Naive Bayes or Support Vector Machines, these methods require intense training with several hundreds or at least dozens of training documents per template to reach their high precision. Updating these learning models with new training documents means a complete learning cycle and is thus hard to achieve in a production environment.

Especially if such automatic document indexing should be used by SOHO[1] users, there is a need to provide an easy to train extraction component, which

[1] Small Office Home Office.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 538–549, 2012.

incorporates feedback immediately as only few training examples are available for each document template. Thus we provide a method which works on instance based document type classification and template detection in combination with layout-based data capture relying on positional OCR[2] results as well as positional user feedback.

This method (including means to limit the size of the model) is the main contribution of this paper and described in Section 4. A thorough evaluation on a real world data set of more than 5,000 business documents has been carried out (Section 5) providing general accuracy measure values as well as showing different aspects like the influence of model size on the extraction quality.

Before this, we start by discussing related work (Section 2) and giving an overview of the solution (Section 3).

2 Related Work

Providing adaptable models for machine learning has already been widely researched. Support Vector Machines (SVM) are one of the preferred methods. Cauwenberghs et al. [2] and Jia et al. [6] demonstrate their classification variability. Incremental and decremental functionality for model entries are shown but without performance evaluation in [2]. Jia processes a derivative method for SVM kernel functions. After a certain time step a new kernel function is generated based on the previous one.

In contrast to Cauwenberghs et al. [2] and Jia et al. [6] we focus on a method that immediately adapts to faulty processed documents and is able to handle index data extraction in addition to document classification. [7,11] propose an active learning method focusing on rich feedback by feature adaptation. Their system is based on features marked by human experts to update the knowledge model of the learning algorithm. Our proposed method is only based on result corrections, which a typical user carries out in his usual workflow anyway. In addition active learning as applied by Raghavan et al. is no update strategy for models but an enhancement strategy for model generation. Similar issues arise with the extraction approach of Culotta et al. [3] where user provided constraints extend a conditional random field method for index field extraction. Our approach assumes a template-based document layout which is exploited for improved extractions.

Stumpf et al. [9,10] evaluated user understanding of classification decisions and methods to adapt the model. Even though they say that users are able to adapt rules and keywords to correct classification results, we require a faster result evaluation. Hence we prefer result correction instead of feature labelling, which is not discussed in [9]. Stumpf exploits the obtained experimental results to implement two methods. The first method uses a constraint-based approach. Relevant feature weights are adapted by increasing and decreasing user decisions. This method shows no improvement for the evaluated scenarios. The second method uses a variant of co-training as presented by Blum et al [1]. Instead

[2] Opitcal Character Recognition.

of using two different classifiers they use one generated from user feedback. Similar to active learning, this method is also restricted to the model generation phase and would need further enhancements to fit into the continuous model adaptation. A similar approach with the same confinement using Naive Bayes classifier and user feedback is given by Huang et. al. [5].

3 Overview

The classification and extraction process as shown in Fig. 1 consists of two main phases. At first a model is generated from training data. The training data provides the correct index fields for each document as well as their positions (bounding box) within the document as our method works layout-based. In the second phase, unkown documents are processed. During processing, each document is at first classified to its document type (e.g., invoice, reminder) before the remaining index fields are extracted based on coordinates from the k most similar documents. Processing is carried out by a k-nearest-neighbour (k-NN) algorithm based on the automatic document type classification and index field extraction from our earlier work [4].

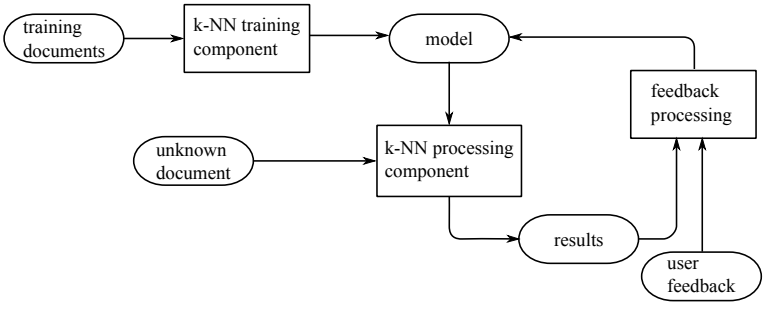


Fig. 1. System overview with main components and data processing

Feature Extraction: Both algorithms (one for classification and one for extraction) consist of an inverted index of documents to generate a model. As shown in [4] we tried several features and feature combinations. In the end the following very simple features showed the best performance and are thus also used throughout the rest of this work. The classification model stores each document under all words from the document and classifies new documents by the type the k documents with the most similar content have. This is a simple word-based retrieval model as used by most search engines. The extraction adds the position of each word to the index and finds index fields based on the location of index fields in the k documents that share the most words at the same position. For this we sliced the document into boxes and appended the box a word starts in to the word in the index. If a document contains the word "invoice" in the upper

left corner for example, it would be stored to the index under "invoice_0_0". The same is true for all other words in the document. That way a query to the index with a new document returns the k documents that have share many words at the same position and thus have a similar layout. Our layout based extraction approach then can apply the coordinates for each index field from the documents in the index to the new documents to extract new values.

Index Data Extraction: Fig. 2 shows an example of several documents using the same template with index fields on similar positions. Thus if the user provides feedback with the correct positions for the first two documents, we are able to find at least the fields sender name and document id as they are at the same position. Surely, this layout-based method does not work for fields with variable positions such as the total amount (as shown in the figure).

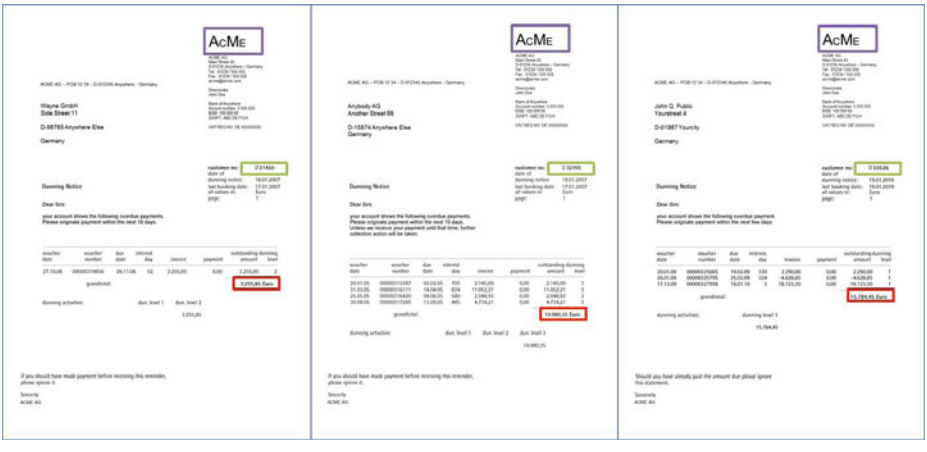


Fig. 2. Business documents with the same template

Real World Scenario: The algorithms described in this paper are used by a mid-sized german company in their document archiving solution. Users access this solution through a web interface. Over this web interface they are able to archive their documents and run automatic classification and indexing. Each user has an inbox of new documents either scanned or loaded directly from hard disk. Index values are assigned to each document with a certain confidence value between 0% and 100% based on the score of the retrieved similar documents from the index. The user can correct these values, thus providing feedback, in two different ways. He can modify the values in the text box or he can click the correct value for a field on an image of the document. These corrections are then sent to the feedback algorithm described in this paper. For simplicity, we concentrate on feedback that is given by clicking on the document's image in the remainder of this paper.

4 Incorporating User Feedback

Feedback processing consists of three main components shown in Fig. 3(a) and discussed in the remaining section. At first, error recognition is explained with three different types of errors. Based on classification and index field errors models are updated and finally less frequently used entries are removed to shrink models.

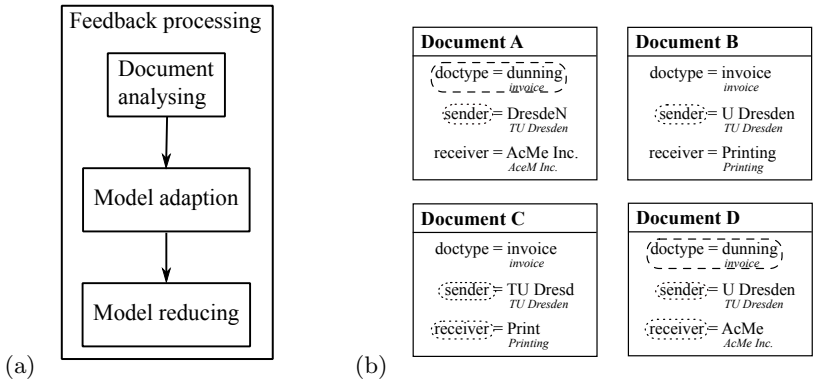


Fig. 3. (a) Feedback processing steps. (b) Faulty processed documents with highlighted grouping. User feedback is marked italic for each extracted value.

4.1 Document Analysis

Feedback processing starts by analysing differences between extracted data and user provided feedback data. User data consist of the corrected document type in case of classification issues and corrected index field with position coordinates. Extraction errors are identified by one of two strategies.

Classification errors are found by checking directly relevant index data and feedback fields. Extraction errors consist of all faulty index fields. Therefore the strategy checks all available extracted and user feedback fields of a document and ignores extraneous fields, e.g., document type or language.

Up to now, three different error types are available. If there was yet no training example available, fields are marked as training error for immediate learning. If classification or extraction delivered wrong results, classification errors and extraction errors are distinguished. Each error is associated with a learning threshold needed for the model adaptation component as described below.

If an error is similar to a previous error, they are grouped together. This way it is possible to consider only groups of errors, thus avoiding overfitting because of outliers. Grouping works as presented in Fig. 3(b). Two classification errors get grouped only if the extracted document type (doctype) is not the same as the user provided one and both expected types are equal. Two extraction errors are grouped if the same index fields are faulty. This happens under the assumption

that the model does not consist of a template with proper positions for these index fields.

The example in Fig. 3(b) groups documents A and D because of the same classification error with equal expected document type. Documents B and C are correctly classified. Extraction errors occur in all documents. Documents A and B as well as C and D are grouped because of similar faulty index fields, as we are grouping by fields and their faulty detected template position and not the extracted value.

4.2 Model Adaptation

After document analysis the feedback process performs model adaptation in two phases.

At first the error groups from the previous document analysis step are checked against a threshold t_e associated to each error type e. Training error groups are immediately added to the model as described in the second phase. If the size of classification or extraction error groups is greater or equal to t_e it is marked for consideration in the second step of model adaptation. For example if $t_e = 2$ and analysis found an error group of size 3, the error represented by that group will be marked for the next step.

In the second phase all marked entries are finally incorporated into the extraction or classification model. For our k-NN-based classifier and extractor this simply means adding the documents with its user-provided correct values to the model. In case of classification, the correct document type is added with all words occurring in the document. The extraction model is extended with a new template containing words with their positions.

4.3 Model Reduction

If documents are only added to the model, it will grow indefinitely. Therefore the last feedback processing step removes old documents, that were not used for some time.

The concept of model reduction is based on a voting mechanism. For this purpose it is necessary to monitor which document or documents from the model are the source of information for classification or extraction. So each time a document is used its votes are incremented.

We assume documents having the fewest votes are the least relevant ones. So models are finally reduced by removing such documents. Three parameters influence model reduction. The maximum model size max_m, the minimum model size min_m and the probation period p. As soon as the model size reaches max_m the feedback process removes $max_m - min_m$ documents from the model. If reduction would simply remove documents with the least amount of votes it would mostly remove all documents, added during the model adaptation phase recently. To avoid this, the probation period p protects documents from reduction until p extractions or classifications using that model took place.

5 Evaluation Results

This section shows the variance over extraction and classification quality, while applying user feedback over time. To measure quality we calculated extraction accuracy per extracted index field. To describe the extraction quality for a document, we calculated the mean of the accuracy for all fields per document.

The following section first introduces the dataset used for evaluation, before showing detailed results and conclusions.

5.1 Dataset

The dataset consists of 5,627 business documents obtained from the archive of a mid-sized German company. Index fields and document types are labelled for all documents by human annotators. The set consists of seven different document types distributed as shown in Fig. 4. This distribution mirrors the expected distribution of business documents in German companies. Each document type includes several different document styles with different layouts. The labelled index fields are shown in Table 1. These are typical index fields for German business documents. However not all of them occur in every document. So a correct extraction (true positive) is measured if the field exists in the document and the extracted value is correct.

5.2 Experimental Setup

The experiments consist of 20 single runs over the whole document set. Each run starts from an empty classification and extraction model. It processes all documents one by one. Which document to process next is based on a random choice. The user feedback is simulated using the human annotated labels. Based on this feedback information, documents are added to the models as described in Section 4. To calculate the current extraction quality, precision and recall are averaged over the 50 last processed documents. For this purpose the extracted index fields are compared with their expected value. Only in case of exact matching the extracted value is accepted as correct (T). If any other value is extracted it takes count as wrong (F), excluding the empty string which leads to a no result (NR) count. All available index fields of the last 50 processed documents are used to obtained results. This leads to measure accuracy as described at Equ. 1. Results for a single run are shown in the lower plots of Fig. 5.

$$accuracy = \frac{\mathrm{T}}{\mathrm{T} + \mathrm{F} + \mathrm{NR}} \tag{1}$$

The system is evaluated with different configurations for the three parameters introduced in Section 4.3. The configurations are listed in Table 2. All three model parameters are varied for classification models as well as for extraction models. Additionally, each of those configurations is tested for different grouping thresholds t (see Section 4.2). At first t was set to 1 to simulate immediate

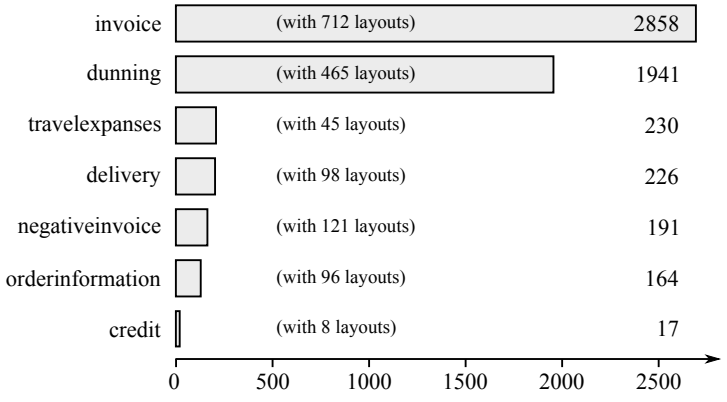


Fig. 4. Document type distribution of the document test set. Each document type contains different layouts listed in the bars.

Table 1. Index fields used for evaluation

Index field
sender
recipient
date
contactperson
tobepaid
docnumber
amount
customerid
contactnumber
email
subject

learning. Then it was set to 3 to research the influence of error grouping on processing performance. The results for each configuration are averaged over 20 runs to remove outliers.

5.3 Representative Performance

Figure 5 shows the performance of one single run with model configuration 1 (See Table 2) and threshold $t = 1$. The upper left side shows results for classification whereas the upper right shows the same for extraction. Obviously both start to grow linearly until their size reaches max_m. This happens after roughly 500 documents are processed. At this point model reduction is invoked and reduces the model to size min_m. Afterwards the model grows linearly until it reaches max_m again and so on. This behaviour continues until all documents are processed. For reasons of space the plots in Fig. 5 are truncated after 3,000 processed documents.

Table 2. Tested model configurations

	max_m	min_m	p
Classification configuration 1	50	40	30
Classification configuration 2	100	80	50
Classification configuration 3	200	180	80
Extraction configuration 1	400	350	70
Extraction configuration 2	1000	800	100
Extraction configuration 3	1500	1300	150

The lower graphics show the model adaptation effect. At first average accuracy rises quickly, staying at a constant high performance after initialization. Interestingly there seems to be no correlation between model reduction and extraction or classification performance. We actually expected performance would drop after reduction, rising again until the next reduction occurs. However the quality variation seems to be a result of random ordering of test documents.

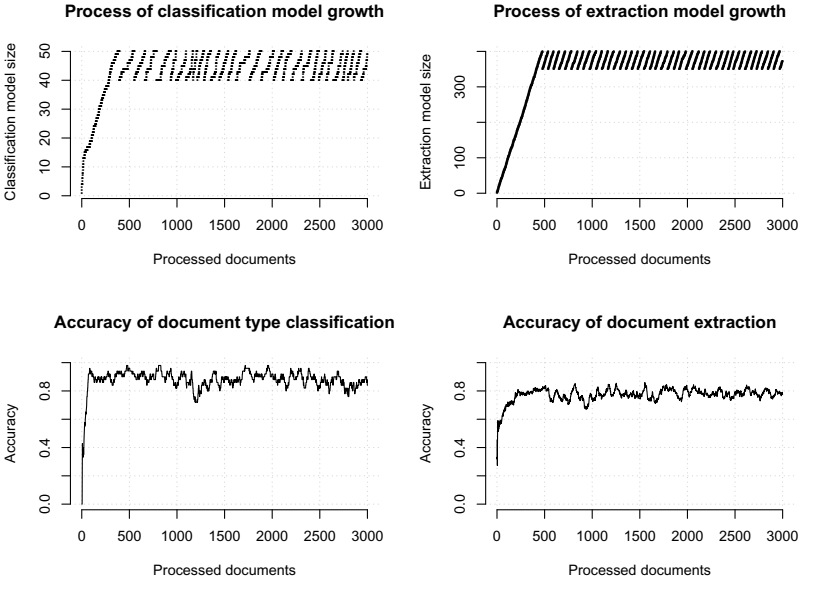


Fig. 5. Evaluation results of one single run with model configuration 1 and immediate learning threshold 1

5.4 Average Performance

The following section explains test results including changes in processing quality as well as runtime performance of our implementation. Table 3 shows the average accuracy for 20 runs and for each of the 3 configurations presented in Table 2.

As expected the results indicate that a larger model leads to better accuracy. This is true because a larger model is able to handle more document variations than a smaller one. However, doubling max_m causes only a slight increase in extraction quality. We can therefore conclude that increasing the model's maximum size is only helpful to a certain extend. Table 3 also shows the influence of different error group size thresholds t. It seems that variations of t have no effect on document type classification but increasing the threshold from 1 to 3 causes a small drop in extraction quality. This might be because classification is less error prone then extraction. While we require a document with the same index values at the same positions for extraction, we require only a document containing similar keywords and having the same type for classification.

Table 3. Average accuracy

	Threshold 1		Threshold 3	
	Classification	Extraction	Classification	Extraction
Configuration 1	0.871	0.775	0.870	0.763
Configuration 2	0.90	0.801	0.899	0.781
Configuration 3	0.924	0.811	0.908	0.785

The graphs in Fig. 5 show a two phase structure. At first the model is initialized with a certain amount of process steps and afterwards shows almost constant behaviour. We investigated this behaviour by counting the amount of processing steps necessary to reach average accuracy of 80%. Table 4 shows the results for the three different configurations presented in Table 2 and the two error group size thresholds. Even though the amount of processing steps for different configurations does not vary much, the influence of the threshold becomes more obvious. A threshold of 3 causes classification to need 50% more processing steps, while extraction takes 3-4 times more steps to build up its model. This was expected since for $t = 3$ an error needs to occur at least 3 times until a document containing the required information is included in the model.

Table 4. Model generation performance. A model is considered to be generated if accuracy hits more than 0.8. Amount of processed documents is shown with proportion of test set.

	Threshold 1		Threshold 3	
	Classification	Extraction	Classification	Extraction
Configuration 1	62 (1.1%)	248 (4.4%)	91 (1.6%)	1271 (22.6%)
Configuration 2	56 (0.9%)	319 (5.6%)	124 (2.2%)	1064 (18.9%)
Configuration 3	65 (1.1%)	345 (6.1%)	139 (2.4%)	1225 (21.8%)

Interestingly, there seems to be no benefit in grouping errors and using a threshold larger than 1. Model generation takes longer and results do not improve. However as Table 5 shows increasing the threshold from 1 to 3 causes our implementation to become 4.9 times faster.

Table 5. Runtime per document of processing and feedback with full set of 5,631 test documents

	Threshold 1		Threshold 3	
	Processing	Feedback	Processing	Feedback
Configuration 1	38.6 ms	15.1 ms	39.3 ms	5.6 ms
Configuration 2	48.3 ms	30.3 ms	44.5 ms	7.2 ms
Configuration 3	53.2 ms	36.7 ms	43.8 ms	7.6 ms

6 Conclusion

We presented an approach to adapt automatic index extraction for business documents to a business' day-to-day situation as well as changing requirements. We also show how to keep indexing quality constant while documents are changing over time. Furthermore, the evaluation shows that even with a fixed maximum model size our method constantly performes with an accuracy of more than 80% using less than 10% of our dataset of real world business documents. Additionally, our model adaptation approach is designed to use data provided by day-to-day users from accounting or management. Therefore, it avoids configuration by an administrator and is usable by small and mid-sized enterprises.

Current challenges include index fields with variable positions and incomplete user feedback. We are going to address these issues in future work.

Acknowledgments. This research and project is / was funded by the German Federal Ministry of Education and Research (BMBF) within the Framework Concept "KMU Innovativ" (fund number 01/S10011A).

References

1. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madisson, WI, USA, pp. 92–100 (1998)
2. Cauwenberghs, G., Poggio, T.: Incremental and decremental support vector machine learning. In: Advances in Neural Information Processing Systems, vol. 13, pp. 409–415. MIT Press (2001)
3. Culotta, A., Kristjansson, T., McCallum, A., Viola, P.: Corrective feedback and persistent learning for information extraction. Artif. Intell. 170, 1101–1122 (2006)
4. Esser, D., Schuster, D., Muthmann, K., Berger, M., Schill, A.: Automatic Indexing of Scanned Documents - a Layout-based Approach. In: Document Recognition and Retrieval XIX (DRR), San Francisco, CA, USA (2012)
5. Huang, Y., Mitchell, T.M.: Text clustering with extended user feedback. In: Proc. of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2006, Seattle, WA, USA, pp. 413–420 (2006)
6. Jia, Y., Yan, S., Zhang, C.: Semi-supervised classification on evolutionary data. In: Proceedings of the 21st International Jont Conference on Artifical intelligence, pp. 1083–1088. Morgan Kaufmann Publishers Inc., San Francisco (2009)

7. Raghavan, H., Madani, O., Jones, R.: Active learning with feedback on features and instances. J. Mach. Learn. Res. 7, 1655–1686 (2006)
8. Saund, E.: Scientific challenges underlying production document processing. In: Document Recognition and Retrieval XVIII, DRR 2011, San Francisco, CA, USA (2011)
9. Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., Drummond, R., Herlocker, J.: Toward harnessing user feedback for machine learning. In: Proceedings of the 12th International Conference on Intelligent User Interfaces, IUI 2007, Honolulu, HI, USA, pp. 82–91 (2007)
10. Stumpf, S., Rajaram, V., Li, L., Wong, W.K., Burnett, M., Dietterich, T., Sullivan, E., Herlocker, J.: Interacting meaningfully with machine learning systems: Three experiments. Int. J. Hum.-Comput. Stud. 67, 639–662 (2009)
11. Wong, W.K., Oberst, I., Das, S., Moore, T., Stumpf, S., McIntosh, K., Burnett, M.: End-user feature labeling: a locally-weighted regression approach. In: Proceedings of the 16th International Conference on Intelligent User Interfaces, IUI 2011, Palo Alto, CA, USA, pp. 115–124 (2011)

Evolutionary Adapted Ensemble for Reoccurring Context

Konrad Jackowski

Departments of Systems and Computer Networks, Wroclaw University of Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
konrad.jackowski@pwr.wroc.pl

Abstract. Reoccurring Context is a phenomenon being subject of interest in machine learning theory dealing with Concept Drift. Periodic reappearance of contexts naturally encourage designing classifier systems which utilizes their expertize on contexts collected in the past. The paper presents study on EAERC algorithm that gather its knowledge on appearing contexts in form of elementary classifiers which can potentially contribute in ensemble classifier system if necessary while keeping ensemble size strictly limited to ensure short response time. While unseen context appears EAERC automatically adds new classifier to the pool.

Keywords: Concept Drift, Multiple classifier systems.

1 Introduction

One of the interesting concepts in Machine Learning theory is designing ensemble classifier as the way of improving classification accuracy in decision making systems. Its strength lies in utilizing of knowledge on a problem being under consideration stored in a pool of elementary classifiers, each of which can participate in decision-making. Regardless of how joint decision-is made one of the basic assumption, made while collecting classifier pool, is necessity of ensuring an adequate level of diversity of classifiers. In other cases additional effort required for processing data does not result in elevating classification accuracy.

When the system works in a static environment, i.e. when the concept does not change in time, diversity may be understood as local competency in feature space, to mention just one option. The situation looks different when the system is designed to work in a dynamic environment, i.e. when variability of concept can be observed in time what is called The Concept Drift. In this case diversity may lies in gathering set of classifiers being experts in recognizing different appearing concepts.

This variability of concept may relate to any of listed below characteristics: a priori probabilities of classes, the conditional probability distribution of features in the classes, posterior probabilities, or even the set of possible class indices [1]. Characteristic of the concept drift in time may also vary. There are couples of types of Concept Drift defined in literature: Sudden Concept Drift, where essential changes of aforementioned characteristics appear suddenly in particular moment of time; Gradual and Incremental Concept Drift, in which the changes occur in an evolutionary manner and

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 550–557, 2012.

are spread in time. In all these cases, the changes are usually assumed to be non-reversible, what means that the previous context is not expected to appear again in the future, although, further changes and possibility of new concepts are not excluded. A separate category is Reoccurring Context. It focuses on the reemergence of contexts that occasionally reappear in time, though the sequence may or, in more general case, may not be known. In all listed concept drift types, the moment of drift is not known.

There are several issues that has to be addressed while designing the ensemble classifier system working under Concept Drift. Among the others we can mention: choice of appropriate moment when a new elementary classifier has to be trained for most recent concept to join the pool; selection of competent classifiers from the pool, which have valuable knowledge and can contribute in decision making of the ensemble and strategy of eliminating outdated classifiers from ensemble.

In the paper the concept of Evolutionary Adapted Ensemble for Reoccurring Context (EAERC) is presented. The algorithm makes the decision according to weighted voting strategy with limited number of voting committee members drawn from pool of context oriented classifiers. Process of ensemble training which aims at selection committee members and setting their weights is treated as optimization problem which is solved using the evolutionary based algorithm aiming at maximizing classification accuracy of the ensemble.

The rest of the paper is organized as follow. Section 2 provides details of ensemble model and its learning algorithm. Section 3 presents results of experimental evaluation of algorithm performance. Section 4 conclude the paper and presents some guidelines for further work.

2 Evolutionary Adopted Ensemble for Concept Drift

There are five assumptions that lie behind concept of EAERC :

1. The EAERC ensemble is assumed to work under reoccurring context and knowledge on the contexts is collected in set of elementary classifiers gathered in the pool;
2. The ensemble makes decision according to weighted voting strategy with fixed and limited number of voting committee members drawn from the pool;
3. Selection of voting committee members and assigning their weights is realized in iterative training process that aims at maximizing ensemble classification accuracy;
4. Training process should automatically recognize if appearing context is the new one or if it has been seen.

Collecting knowledge on emerging contexts is typical feature in ensemble based solutions, nonetheless as with passage of time some classifiers become outdated it does not make sense to utilize all of them for decision making. In practice some algorithm of substitution of outdated classifiers with new one trained on the most recent context can be applied. The drawback of this approach is that usually rejection of the committee member is permanent, i.e. we lose the knowledge of rejected classifier that could

be utilize in case of reemerging of the context, as it, by assumption, takes place in reoccurring context problems. To avoid the necessity of training new classifier for previously processed context again, EAERC permanently store all classifiers in the pool. Providing that the set of possible contexts emerging in time is finite, it can be also assumed that the size of the pool will be also limited and the process of its updating will end together with processing all of possible concepts.

On the other hand we have to keep in mind that creating voting committee with to large number of classifiers causes extending processing time. The reasonable solution then is limiting ensemble size and creating the committee by means of selected from the pool classifiers. That approach, apart from keeping the processing time during recognition phase limited, can significantly reduce the risk of decreasing ensemble accuracy by contribution of outdated committee members.

Therefore the committee size is strictly limited in EAERC and only selected elements from the pool join the ensemble while the rest waits for the chance in the future. Number of Voting committee members is one of the major EAERC parameter which has to be set arbitrarily.

Other method of elevating accuracy of the ensemble is weighted fusion of discriminating function of the committee members, instead of simple fusion of their decisions. Weights can control the level of contribution of particular classifier in the ensemble decision making and intuitively should be straight proportional to expertise level of respective classifier. There are number of proposition how to set the weights including implementing forgetting algorithm, in which the weight is counter proportional to time when respective classifier was trained, or others which compute the weights proportionally to classifier accuracy evaluated for currently processed contexts. In EAERC it has been decided that process of weights calculation will be treated as ensemble optimization process that aims at maximizing its classification accuracy over current context.

The last issue to deal with is selecting the moment when the pool should be updated with a new classifier. Common practice in ensemble solution is launching the procedure after processing some chunk of stream data without dedicated trigger procedure. That approach works well if past context does not happen to appear in the future and creating new classifier coincide with rejecting the other one from committee, but it does not make sense when reoccurring context is assumed. Therefore some trigger procedure is proposed in EAERC that is fired in two cases:

1. when the size of pool is smaller than maximal committee size. That option will be fired at the beginning of training process when the first chunks of data stream are processed until the pool has enough elements to fill the voting committee;
2. when evaluating ensemble accuracy decreases while processing subsequent data chunk. That situation is likely to happen when new context appear and EAERC training procedure cannot find appropriate experts in the pool.

All of above presented considerations allow us to define the following ensemble decision making formula:

$$\overline{\Psi}(x) = i \Leftrightarrow \arg\max_{i=1}^{M} \sum_{\Psi_k \in \Pi^{\Psi}}^{K} w_{k,} \delta\left(\Psi_k(x), i\right). \tag{1}$$

Where $\overline{\Psi}$ is the ensemble classifier,

δ is Cronecker's delta,

$w_{k,}$ states for weight of k th committee member,

Ψ_k denotes k-th committee member, and

Π^{Ψ} denotes voting committee.

As mentioned before EAERC training procedure aims at maximizing classification accuracy of the ensemble over currently processed data chunk. The goal can be achieved by searching set of possible committee configuration by manipulating with two parameters:

- selection committee members from the pool Π^{Ψ},
- modification of the weighting factors $w_{k,}$

EAERC employs for that purpose tailored evolutionary based training algorithm which processes population of individuals representing possible solution encoded in a form of compound chromosomes consisting of two constituents (Eq. 2):

- vector of indexes of classifier in the pool drawn to join the committee,
- vector if weights assigned to committee members.

$$Ch = \begin{cases} [I_1, I_2, ..., I_K] \\ [w_1, w_2, ..., w_K] \end{cases} \tag{2}$$

where I_k states for classifier index in the pool related with k-th committee member.

EAERC training procedure is an iterative process of entire population transformation with means of crossover and mutation genetic operators. The first one is a standard two-point crossover operator with two parent and two offspring which affects both chromosome constituents.

Mutation operator is somewhat more complex and affects both chromosome constituents in different manner. Mutation of weights vector is realized by adding some random noise generated with Gaussian distribution. The indexes vector is affected by substitution on its randomly selected constituents with the indexes of other classifier from the pool keeping constraint that particular classifier from the pool can be used in the committee only once.

The quality of solution represented by particular individual is measures using fitness function that calculate number of correctly classified samples stored in training set.

Pseudo code of the proposed algorithm is given below

```
Start
Load training set
Initiate algorithm
  begin
  Set Chunk Size
  Set Maximal Size of Voting Committee;
  Set population size;
  Set mutation rate;
  end
Initiate population
Repeat until end of training set
  begin
  Get next Data Chunk from training set;
  Repeat until last generation
    begin
    If (Update Classifier Pool Criteria)
    begin
    Train New classifier over Data Chunk;
    Add New classifier to the pool;
    Add New classifier to randomly drown individuals;
    end
  Evaluate population over the data chunk;
  Select elite from population
  Reproduction
  Mutate part of population
  Crossover rest of population
  Create child population
  Save population;
  Save best result;
  end
Select and return best individual;
End
```

An additional words of explanation on criteria of updating the pool with new classifier. In practice the fitness of the best individuals in the population is saved and compared with the results of the winner in subsequent iteration. If the decreasing of the fitness is noticed the procedure of creating new classifier is lunched.

3 Experiments

This chapter will present the results of experiments, carried out in order to evaluate the performance of EAERC. Since the paper presents basic concept of the proposed algorithm and the algorithm is still under construction, only basic comparative

performance analysis is provided with two optional recognition algorithms: Single Best (SB) and Majority Voting committee 3SBMV. The first one made its decision with means of one single classifier drawn from the pool of all trained elementary classifier which had gained the highest classification accuracy over the actually processed data chunk. The second one (3SBMV) made an collective decision according to majority voting of three best classifier drawn from the pool.

We limited experiments to two artificially generated benchmark datasets. Both of them consisted of 100.000 objects with features in two-dimensional spaces generated according to Highleyman and Banana distribution respectively. In order to simulate the Concept Drift, the sets were divided into subsets. Features of instances in subsequent subsets were rotated at 90 degrees in feature space what resulted in creating forth concepts, which we named A,B,C,D. To simulate reoccurring context, each concept appeared in the dataset twice.

Entire data stream was split into set of 32 data chunks, four chunks per each context presentation. Each data chunk was then randomly spited into three datasets of equal number of instances. The first one was used for training elementary classifiers. The second one was input data for training EAERC algorithm and selecting elementary classifiers exploit by SB and 3SBMV algorithms. The last one was used for the purpose of classifier evaluation.

As the EAERC algorithm is random by its nature (random generation of initial population), the experiments for each data set was repeated ten times and obtained results was averaged.

Maximal size of voting committee was limited to three.

Experimental environment was implemented in Matlab using PRTools library and Genetic Toolbox. Simple back propagation neural networks were used to create elementary classifiers.

Results of experiments are summarized in Figure 1 and Figure 2

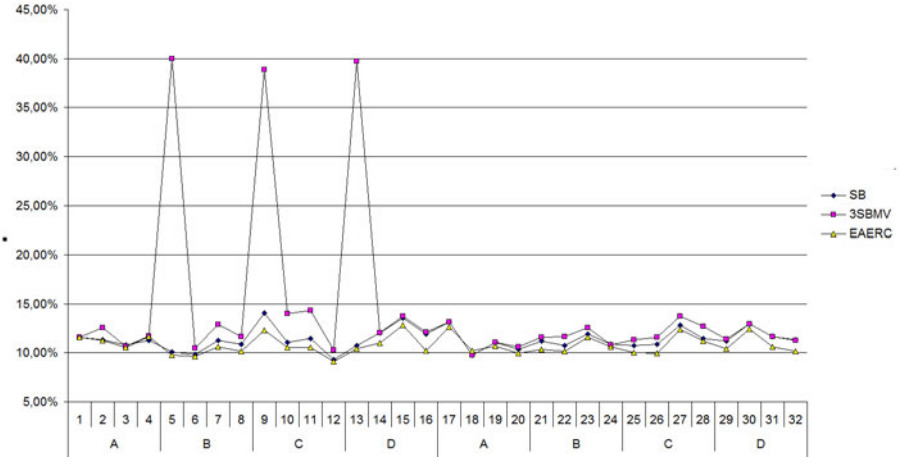


Fig. 1. Misclassification rate of EAERC for Highleyman dataset

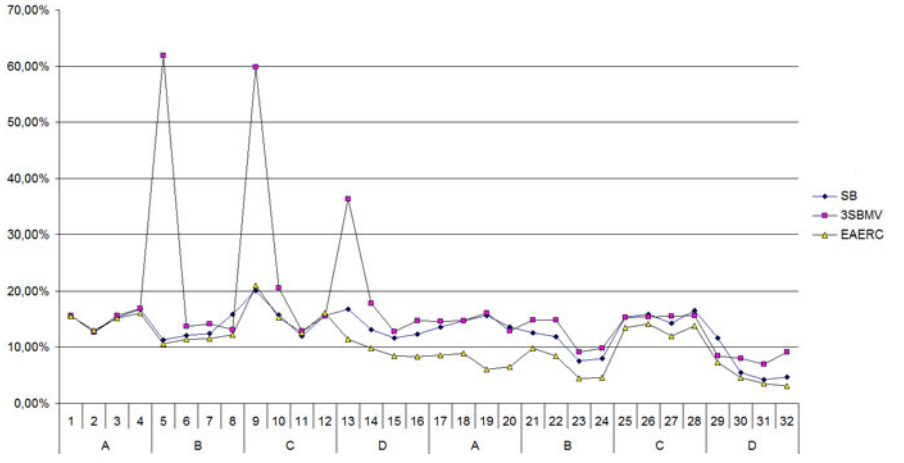


Fig. 2. Misclassification rate of EAERC for Banana dataset

For both datasets following common observation can be done:

1. In both experiments EAERC outperforms other classifiers
2. Emerging of new concepts noticed at step 5, 9, and 13 result in sudden raise of misclassification rate of simple majority voting classifier (3SBMV) which is then gradually reduced within next 2 steps;
3. No significant changes in performance of EAERC classifier can be noticed when new concept emerges.
4. Reoccurring of the concepts that appear at steps 17, 21, 25 and 29 steps affect the misclassification rate in much smaller degree.

The first observation profess the ability of the EAERC training algorithm to effectively select the competent committee members form the pool of classifier and to set their weights respectively to their expertise level according to presently valid context.
The emergence of a new context that has not appeared previously (observation 2) causes dramatic lowering of quality of the 3SBMV because the committee is dominated by classifiers trained for another context. Processing each subsequent chunk and creating next elementary classifiers fill the committee with majority of experts on given context that lead to improving the performance.

At the same moments EAERC algorithm keeps its performance on the same level (observation 3). That shows that ability of weighting committee members voice in decision making can be effective way of elimination destructive contribution of outdated classifiers and that EAERC algorithm sets the weights appropriately.

Lack of dramatic changes in performance starting form 17 step regardless context drift is the result of utilizing frozen classifier from the pool.

The presented results seem to be optimistic and encourage further works that shall consist of:

1. Evaluating EAERC over larger number of benchmark datasets,
2. Carrying on comparative analysis of EAERC with alternative ensemble based solutions.

4 Conclusions

The paper presents a new Evolutionary Adapted Ensemble for Reoccurring Context algorithm dedicated for decision making systems working under reoccurring context. The algorithm gathers knowledge on all emerging contexts while keeping voting committee size limited. The algorithm has built in procedure that automatically triggers creating new classifier when new context emerges. Presents the results of preliminary experiments on EAERC performance encourage for further work on the algorithm.

This work is supported in part by The Ministry of Science and Higher Education under the grant which is being realized in years 2010-2013.

References

1. Kelly, M.G., Hand, D.J., Adams, N.M.: The impact of changing populations on classifier performance. In: KDD, pp. 367–371 (1999)
2. Littlestone, N., Warmuth, M.K.: The weighted majority algorithm. Inf. Comput. 108(2), 212–261 (1994)
3. Oza, N.C.: Online ensemble learning. In: AAAI/IAAI, p. 1109. AAAI Press, The MIT Press (2000)
4. Breiman, L.: Pasting small votes for classification in large databases and on-line. Machine Learning 36(1-2), 85–103 (1999)
5. Wang, H., Fan, W., Yu, P.S., Jiawei, H.: Mining concept-drifting data streams using ensemble classifiers. In: Getoor, L., Senator, T.E., Domingos, P., Faloutsos, C. (eds.) KDD, pp. 226–235. ACM (2003)
6. Tsymbal, A., Pechenizkiy, M., Cunningham, P., Puuronen, S.: Dynamic integration of classifiers for handling concept drift. Information Fusion 9(1), 56–68 (2008)
7. Zliobaite, I.: Learning under concept drift: an overview.Technical report, Vilnius University, Faculty of Mathematics and Informatic (2009)
8. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning, 69–101 (1996)
9. Wozniak, M.: Experiments with trained and untrained fusers. AISC, vol. 44, pp. 144–150 (2007)
10. Jackowski, K., Wozniak, M.: Method of classifier selection using the genetic approach. Expert Systems 27(2), 114–128 (2010)

Drift Detection and Model Selection Algorithms: Concept and Experimental Evaluation

Piotr Cal and Michał Woźniak

Wrocław University of Technology,
Department of Systems and Computer Networks,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{piotr.cal,michal.wozniak}@pwr.wroc.pl

Abstract. The traditional classifier cannot keep its quality, when the concept drift appears. The paper proposes how to protect against classification quality decreasing when concept drift occurs. Invented methods do not train classifiers all the time but they try to use earlier gained knowledge about models and switched older model to suitable new one. In this work we assume that the set of models is known and stored as the pool of classifiers. Then, by using drift detecting and searching models methods, we can choose the best model. Our propositions and the main characteristics of them were evaluated on the basis of the experiments which were carried out on chosen artificial data set.

Keywords: Machine learning, supervised learning, concept drift, pattern recognition.

1 Introduction

Rapid progress in computer science causes that machine learning is often used to solve many contemporary problems. But in reality, the majority of humans or industry preferences are dependent on time, place, or another factors. For example, the stock exchange behaves differently in global crisis times and needs new models to predict behavior, but when situation starts to stabilize, "old" knowledge could be useful again.

The majority of intelligent systems are designed on the basis of supervised learning method, what means that a given examples are labeled by experts. That is very useful for pattern recognition or classification problems which aim is to find a correct class on the basis of observed characteristics of a given object [4].

In this work we will not focus on classifiers design. The main aim of the work is to propose the methods of drift detecting and model switching. Concept drift problem is not a new one but nowadays it is a focus of intense research. However, its analysis has not been done completely and many problems are still remain unsolved. Usually, algorithms, which are able to deal with concept drift, are equipped with drift detection system. Otherwise, algorithm must constantly learn to keep its efficient [5], [8]. However, the process of learning needs some resources, time and, especially, learning examples which could be hard to gather

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 558–568, 2012.

and label [12]. System with drift detector can adapt to drift only when situation needs it. This way could be better and more effective for some kinds of drifts (e.g., spare time or resources) than constant learning [3], [10].

During supervised learning we need a feedback from users to predict changes. In this work we use classification errors which could protect us against decreasing of classification quality and it is helpful tool to predict concept drift appearance. Nevertheless, in the real problem it is usually impossible to design perfect classifier. We have to take into consideration that error committing is a part of a typical decision support systems activity and detecting methods should factor this observation in.

Nowadays the access to huge memory resources is not as problematic and expensive as it used to be. In this way, once learned model (classifier for a special model) could be stored and possibly used in future. If the set of models is big enough and the concept drift occurs, we need to search for an appropriate model and switch between old and new one. If the set is small, then probably we cannot find satisfied model and we need to train it. This way could be useful in an unstable environment where changes are common, especially when recurring concept drift takes place.

In known research papers, we rarely can find system with models' switching, which is tested in this work. This solution seems to very effective in recurring drift case [6]. This method needs a detection system, a switching algorithms, and a set of models already learned.

The paper proposes novel methods of drift detection and searching for an appropriate model. The quality of proposed methods was evaluated on the basis of computer experiment which was carried out on the basis of the artificial dataset.

2 Problem Statement

Let us formalize classification task. We would like to assign a given object describe by feature vector $x = [x^{(1)}, x^{(2)}, \ldots, x^{(d)}]^T \in X \subseteq R^d$ to the one of the predefined class ω_i, where $\omega_i \in \Omega = \{\omega_1, \omega_2, \ldots, \omega_k\}$. For inductive learning a classifier is constructed on the basis of learning set $LS = \{(x_1, \omega^{(1)}) \ldots (x_N, \omega^{(N)})\}$, where x_i denotes feature vector of the ith object and $\omega^{(i)}$ is its label respectively. The aim of the classification is to construct such a function $\psi : X \to \Omega$ which allows to assign each observed object to one of the class from Ω [7], [2].

Many useful classifiers have been developed till this time [7], [2], [1], but they usually assume that classifier model is stable. In many practical areas of interest a concept could change depending on hidden context [8] as time, places, or environment. This changes are named "concept drift". It leads to the situation that the model used by classifier become inappropriate. Desired system should adapt to changes in such a way that his quality will not be decreasing or the decrease will be minimalized.

$P(x_t, \omega_i)$ denotes the probability of object x belonging to class ω_i in t moment. We can observe concept drift if $P(x_{t+1}, \omega_i) \neq P(x_t, \omega_i)$. Drift has few types like

"sudden" – when noticeable changes appear instantly, or "gradual" – when little changes have become [9].

3 Algorithms

3.1 Drift Detector

Drift detector is an important part of switching models system. It gives information about moment when model change is needed. We propose to use information about errors from "users" response and we could suppose that concept drift occurs on the basis of mean error and standard deviation from a subset of following objects, which is usually called window. However, to eliminate strong mean error's oscillation causes by irregular distribution of classification errors we count mean again from window but this time from earlier mean error, what is shown in fig. 1. When drift occurs, almost ever, error rate starts growing . Idea of this detection method could be found in [3], where errors and standard deviation are used to predict concept drift. Authors of mention above work use support for a decision, but we propose to use typical classification error. If x_n represents one sample, then $d(x_n) = \omega_n$ denotes classifier decision output, where ω_n denotes class label. If ω'_n denotes expert decision and *window* denotes window size, then the nth mean error and standard deviation could be formulated as:

$$err_{1mean_n} = \frac{\Sigma_{i=n-window}^{n} Id(x_i)}{window} \tag{1}$$

where $Id(x_n) = 0$ if $d(x_n) = \omega'_n$ or $Id(x_n) = 1$ if $d(x_n) \neq \omega'_n$

$$err_{2mean_n} = \frac{\Sigma_{i=n-window}^{n} err_{1mean_i}}{window} \tag{2}$$

$$std_n = \sqrt{\frac{\Sigma_{i=n-window}^{n} (err_{1mean_i} - err_{2mean_n})^2}{window - 1}} \tag{3}$$

Algorithm 1 shows the pseudocode of drift detector method. When classification process on the test set is going on, the algorithm computes err_{2mean_n} and std_n. They are stored if their sum is minimal. If the sum is greater than the sum of the stored mean error and β-fold standard deviation, which could be interpreted as the level of permitted errors, then detector supposes concept drift and runs algorithm responsible for the correct model searching. The parameter β should be selected by parametric analysis for an individual problem.

3.2 "Compare all Models" Method

The first algorithm of a new model searching after drift detection tests all models in a given model set by some period of time, when all available classifiers are trying to recognize a given set of examples (the number of examples is algorithm parameter). Then the model with the lowest classification errors is chosen.

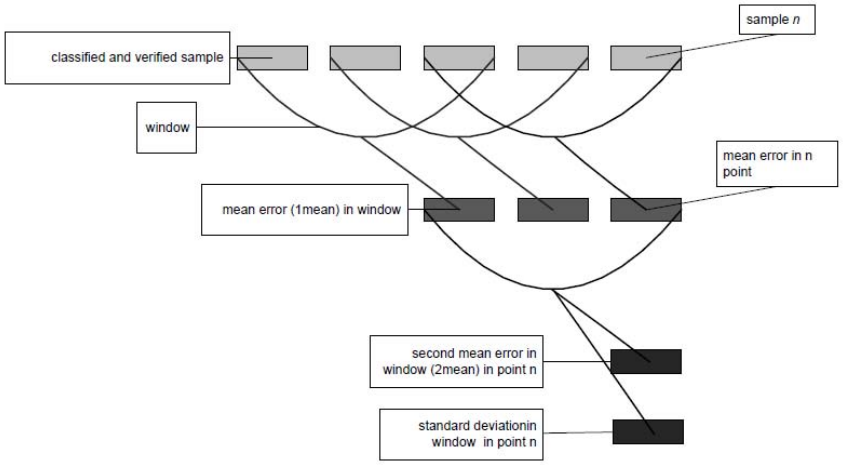


Fig. 1. Example about gathering information about errors by detector

Algorithm 1. Drift detection algorithm

Require: $err_{min} = 1$ and $std_{min} = 1$ (always set when model change)
  for n sample **do**
    compute err_{2mean_n} and std_n
    if $err_{2mean_n} + std_n < err_{min} + std_{min}$ **then**
      $err_{min} = err_{2mean_n}$ and $std_{min} = std_n$
    end if
    if $err_{2mean_n} + std_n$ ¡ $err_{min} + \beta \cdot std_{min}$ **then**
      run searching model process
    end if
  end for

During a searching process, class labels are chosen according to the model with actual the lowest number of errors. It means that classifier tries to use as good as possible model. It is very simple and effective method, however, when the set of models is very large then an exhausting search in a short time is difficult. Algorithm 2 shows the pseudocode of this method.

3.3 "Random Searching" Method

The second algorithm is more convenient for the large set of models but less precise. This method is dedicated for classifiers ensemble only. It draws models of ensemble according to a given probability. In the successive steps the probability of models drawing is changing according to its classification error.The algorithm stops, when one model is drawn to all individual classifiers and this model wins. In this way models which often make mistake are omitted and good ones are

Algorithm 2. Compare all models

```
Require: M – array of models
  ERR – array of models errors
  C – array of classifiers decisions according to a model
  limit – number of tested sample set by user
  Function classify returns the decision of a given classifier for a given object.
  for n to n+limit sample do
        tmp = min(ERR)
        for k = 1 to size(M) do
              C[k] = classify(the nth sample,M[k])
              if tmp = k then
                    classifier decision is a sample class label
              end if
        end for
        check user response
        for k = 1 to size(M) do
              if C[k] classified incorrectly then
                    ERR[k] = ERR[k]+1
              end if
        end for
  end for
  return M[min(ERR)]
```

favored. The algorithm works better when classifiers ensemble has a sufficient number of single classifiers at its disposal. Algorithm 3 shows the pseudocode for this method.

4 Experiment

The main goal of this experiment was to establish the influences of the proposed algorithms on the quality of classification and models selection.

4.1 Set–Up

Data Set. All experiments were carried out on the benchmark dataset describes in [5] which is called SEA. Each object belongs to the on of two classes and is described by 3 numeric attributes with value between 0 and 10, but only two of them are relevant. Object belongs to class 1 (TRUE) if $att_1+att_2 < \phi$ and to class 2 (FALSE) if $att_1 + att_2 \geq \phi$. ϕ is a threshold between two classes, so different thresholds correspond to different concepts (models).Thus, all generated dataset is linearly separable, but we add noise, which means that class label for some samples is changed, with expected value equal to 0. The number of objects, noise and the set of concepts are set by user.

Methodology. We simulated drift by instant model change which means that after some number of samples the decision boundary (ϕ correspond to model) are

Algorithm 3. Random searching method

Require: M – array of models
CLAS – array of classifiers in ensemble
RM – array for storing random models
P – array of probability of selecting models
C – array for storing individual classifier decision according to random model
Function *classify* returns the decision of a given individual classifier for a given object according to random model.
enhance; reduce – parameters set by user
while n sample **do**
  random to RM models from M with probability P
  if all models in RM are the same **then**
    return RM[1]
  end if
  for $k = 1$ to *size*(CLAS) **do**
    C[k] = *classify*(the nth sample,CLAS[k],RM[k])
  end for
  compute ensemble decision
  check user response
  for $k = 1$ to *size*(C) **do**
    if C[k] classified correctly **then**
      P[RM[k]] = enhance·P[RM[k]]
    else
      P[RM[k]] = reduce·P[RM[k]]
    end if
  end for
end while

changed. The sequence of models describe switching from one model to another. Correctness was counted for a classification system with our algorithms and when we had optimal model for every sample. All tests were carried out on ensemble of classifiers denotes by "en" [7] which contained: decision tree, naive Bayes, k-NN, linear and quadratic discriminant analysis [1], [2], [7]. Searching models algorithms denote by:

- "compare all models" method – "CAM"
- "random searching" method – "RSM"
- reference method – "auto" (when the optimal model is used to every sample)

The sequence of models (the set of models) we mark in parenthesis "<", ">". Fig. 2 shows the stages of drift detector working.

Evaluation Methods. The dataset was divided into two equal parts. The first part was for training classifiers and the second one was used for testing and evaluation. To measure the quality of the drift detector we compute the difference between accuracy of a given classifier using ideal drift detector (denotes as "auto") and accuracy of this classifier using tested drift detector.

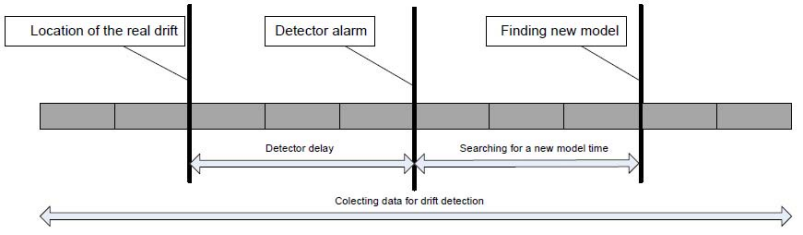


Fig. 2. Stages of drift detector working

Because our artificial data set was numeric we could calculate difference between models. Two significant features had values between 0 and 10. So maximum sum could be 20 and minimum 0. If our model is $< 0 >$ ($\phi = 0$), then all samples were classified as FALSE and for $< 20 >$ as TRUE respectively, what designated maximum value of change. Let m_F denotes model $< 0 >$ and m_T denotes model $< 20 >$, m_O denotes the optimal model for classification, and m_C denotes model which is chosen by algorithm.

$$f(m_O, m_C) = |m_O - m_C| \cdot \frac{|1}{|m_F - m_T|} \tag{4}$$

So, $f(m_O, m_C = m_O) = 0$,
and $f(m_O = m_T, m_C = m_F) = 1$ or $f(m_O = m_F, m_C = m_T) = 1$

In simulation all samples were rated and averaged.

$$Rate = \frac{\Sigma_{i=1}^{n} f(m_O, m_C)}{n} \tag{5}$$

This method of rating is appropriate when we used "CAM" algorithm, because all samples were classified by using one model. In "RSM" algorithm, in a time when it was looking for a new model, it was not possible to define which model was used for classification. In our research we did not use this samples to compute rate. This way could promote algorithm but the rate gain was very small so it could be omitted.

4.2 Results

Fig. 3 shows how classifiers ensemble tries to follow concept drift when changes are significant. Despite 10 percent of noise, the methods quick and very precise find the new appropriate model. In fig. 4 we can see opposite situation when changes are insignificant and noise level is greater than level of change. In this case algorithms do not select the optimal model but often close to the optimal and they try keep level of correctness. The detector detects concept drift worse, because number of errors caused by drift are less and it does not "know" if errors are part of the system activity or they are caused by concept drift. In both cases detector detects drift despite the optimal model was used. The reason is

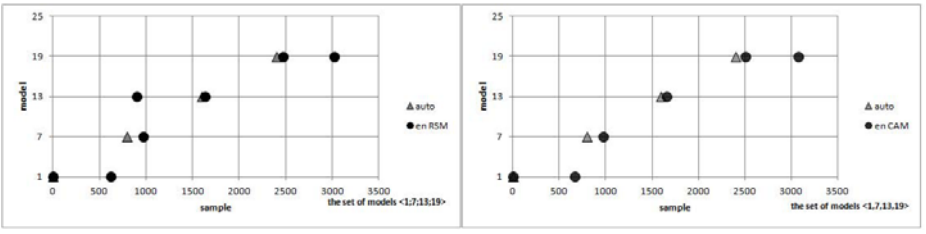


Fig. 3. Selected model and moment of selection. Initial value: 10 percent of noise, 3200 samples, β =4, window size=200,CAM parameter limit=50, RSM parameters enhance=1.05 and reduce=0.8, the set of models $< 1; 7; 14; 19 >$.

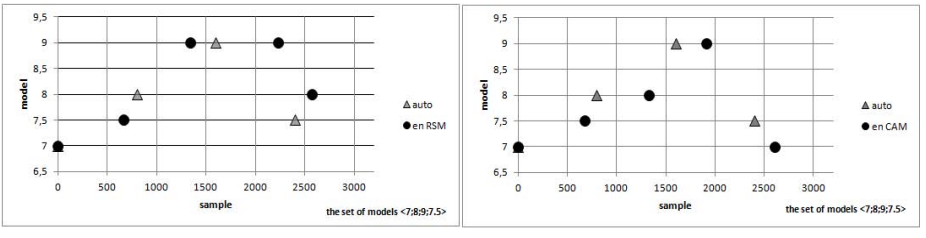


Fig. 4. Selected model and moment of selection. Initial value: 10 percent of noise, 3200 samples, β =4, window size=200, CAM parameter limit=50, RSM parameters enhance=1.05 and reduce=0.8, the set of models $< 7; 8; 9; 7.5 >$.

irregular distribution of errors which are caused by noise and classifier mistakes. Algorithms' parameters were set arbitrarily but they values were established on the basis of the expert's model analysis.

Fig. 5 shows tracking concept drift when small changes go constantly. They present results of experiments when 10 percent of noise were added (in the graph no. 1 and 3) and only 1 percent (in the graph no. 2 and 4). For "RSM" in the first graph (no.3) the parameters of enhance and reduce are very high. It causes that selected models are often very far from the optimal. In the second graph (no. 4), when parameters are chosen wisely, algorithm is a much more better.

The fig. 6 shows how the rate reflects on the correctness decreasing. For both algorithms and for four tests the ranking of rate and the decrease of correctness are very similar. The rate shows how algorithms choose models and it does not look on classification. However, if model is incorrect then quality of classification is also lower.

4.3 Remarks

The main problem of detector methods is error distribution, because it is often irregular. Even if parameters are chosen wisely detector sometimes may alarm

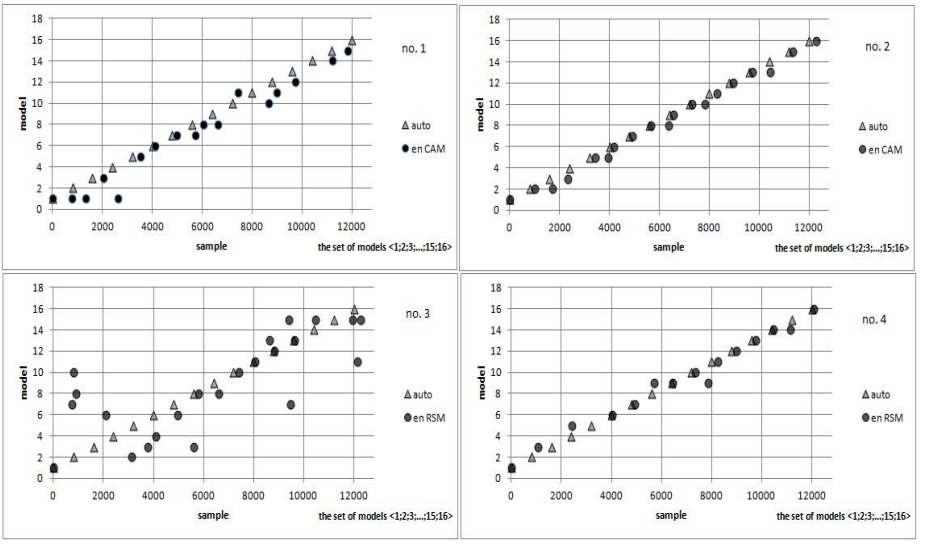


Fig. 5. Selected model and moment of selection. Initial value: 12800 samples, β =4, window size=200, the set of models $< 1; 2; \ldots; 16 >$. CAM parameter limit=50, 10 percent of noise in graph no. 1 and 3, 1 percent of noise in graph no. 2 and 4, RSM parameters enhance=1.4 and reduce=0.7 in the graph no. 3 and enhance=1.05 and reduce=0.8 in the graph no. 4.

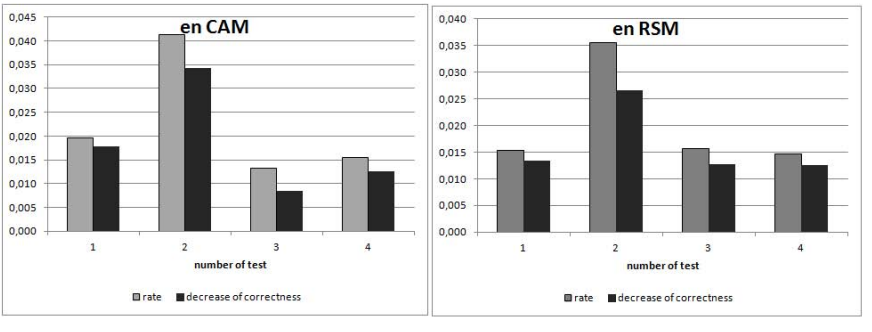


Fig. 6. Rate and the decrease of correctness for four tests. Initial value: 10 percent of noise, 3200 data, β =4, window size=200, CAM parameter limit=50, RSM parameters enhance=1.05 and reduce=0.8 , the set of models $< 7; 8; 9; 7.5 >$.

about drift when the optimal model is used. CAM algorithm is very simple and effective in selecting new model. The method has only one parameter and its selecting does not make trouble. Disadvantage of the CAM is time of searching when the set of models is huge but we have to emphasize that it does not need too many samples to search for a model and it is very accurate.

RSM algorithm has more troublesome parameters. Good choice of them can speed up or increase precision of a selected model. This algorithm works better when it has a large set of models at its disposal. The reason why we set the asymmetric value of parameters is that in our dataset it is much more chance that a single classifier with an insufficient model classify a given sample correct then a classier with fine model classified wrong.

The proposed rate (5) shows how far from optimal model is a given detection and selection algorithm. It is good method to evaluate algorithms of model searching. The value of the rate often reflects the decrease of correctness. However, we can use this rating methods only if it is possible to measure distance between the models.

5 Final Remarks

The decrease of classification quality causes by concept drift is minimized when we use detecting and searching algorithms. Despite unstable environment and another factors which make classification problematic, methods follow the concept drift and they cause classification process more stable. Advantages and disadvantages of proposed algorithms were discovered during the experiments, which showed that switching models method is quite efficient solution and lead to increasing a system's performance.

The propose methods need to have the set of models which has to be learnt earlier. Algorithm equipped with the possibility of learning, storing and switching models seems to be very interesting solution to deal with concept drift problems. Such a algorithms could constantly gather knowledge by some period of time and it would be resistant to changes.

We realize that the scope of experiment was limited and all experiments were carried out on artificial data. Therefore our future work will be focused on more detailed experiments evaluations of proposed methods on real datasets.

Acknowledgment. This work is supported in part by The Polish Ministry of Science and Higher Education under the grant which is realizing in years 2010-2013.

References

1. Alpaydin, E.: Introduction to Machine Learning. MIT Press, Cambridge (2010)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with Drift Detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
4. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
5. Nick Street, W., 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 (KDD 2001), pp. 377–382. ACM, New York (2001)

6. Sobolewski, P., Woźniak, M.: Artificial Recurrence for Classification of Streaming Data with Concept Shift. In: Bouchachia, A. (ed.) ICAIS 2011. LNCS, vol. 6943, pp. 76–87. Springer, Heidelberg (2011)
7. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. J. Wiley, Hoboken (2004)
8. Gerhard, W., Kubat, M.: Learning in the Presence of Concept Drift and Hidden Contexts. Vienna: Ö sterr. Forschunginst. fü r Artificial Intelligence (1994)
9. Zliobaite, I.: Learning under Concept Drift: an Overview. CoRR (2010)
10. Baena-García, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavaldà, R., Morales-Bueno, R.: Early Drift Detection Method (2006)
11. Anton, D., Ulrich, R.: Adaptive concept drift detection. Statistical Analysis and Data Mining 2(5-6), 311–327 (2009)
12. Kurlej, B., Wozniak, M.: Active learning approach to concept drift problem. Logic Journal of the IGPL (2011), doi:10.1093/jigpal/jzr011
13. Li, P., Wu, X., Hu, X.: Mining Recurring Concept Drifts with Limited Labeled Streaming Data. In: 2nd Asian Conference on Machine Learning, ACML 2010 (2010)
14. Kuncheva, L.I.: Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. In: 2nd Workshop SUEMA 2008 (ECAI 2008), pp. 5–10 (2008)

Decomposition of Classification Task with Selection of Classifiers on the Medical Diagnosis Example

Robert Burduk and Marcin Zmyślony

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

Abstract. The article presents the concept of decomposition of the multidimensional classification task. The recognition procedure is divided into independent blocks. These blocks can be interpreted as lower classification problems. The structure of these blocks is presented as a decision tree. In this model the experts give the decision tree structure. The problem discussed in the work shows a selection of different classifiers (or their parameters) to the internal nodes of the decision tree. Experiments conducted for selected medical diagnosis problem show that the use of different classifiers can improve the quality of classification.

Keywords: Hierarchical classifier, error probability, ensemble classifiers.

1 Introduction

The basic idea involved in a multistage (sequential) approach is to break up a complex decision into a collection of several simpler decisions [1-3]. The multistage pattern recognition has two approaches: a decision tree classifier and the hierarchical classifier. The decision tree classifiers organized a series of test questions and conditions in a tree structure [4]. This approach built the classification model and their tree structure is built in the learning process. Hierarchical classifiers are a special type of multistage classifiers which allows rejection of class labels at intermediate stages. The synthesis of the hierarchical classifier is a complex problem. It involves specification of the following components [5]:

1) design of a decision tree structure,
2) selection of features used at each non-terminal node of decision tree,
3) choice of decision rules for performing the classification.

In this paper we will present a combination of both approaches. The computer recognition model is based on a hierarchical classifier. However, in the internal nodes we are using different classifiers, including decision trees, neutral networks, k-NN and ensemble classifiers. Additionally, the design of a decision tree structure in our approach to the hierarchical classifier is based on human expert knowledge. This means that the decision tree structure is not generated automatically, but is presented by an expert.

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 569–577, 2012.

The concept of a decision support system for diagnosis in acute abdominal pain is not new. Many papers are corresponded to this diagnosis problem. One of the first applications in the field of diagnosis of acute abdominal pain was presented in the work [6]. The rule-based learning systems [7-8] and other approaches [9-10] are well suited for the diagnosis in acute abdominal pain. This paper focuses on using different classifiers (or their parameters) in each node of the decision tree problem.

The content of the work is as follows. Section 2 introduces the idea of the hierarchical classifier. In the next section we describe a mathematical model of the acute abdominal pain decision problem and we present results of the experimental investigations of the proposed decision tree. The last section concludes the paper.

2 Hierarchical Classifier

In our consideration the decision rules are based on the probabilistic approach to pattern recognition. This approach consists in the assumption [11] that the feature vector $x = (x^{(1)}, x^{(2)}, ..., x^{(d)})$ (describing the object being under recognition) and the class label $j \in \{1, 2..., M\}$ (the object belonged to) are the realization of the pair of the random variables X, J. Random variable J is described by the prior probability p_j, where

$$p_j = P(J = j). \tag{1}$$

X has the probability density function

$$f(X = x \mid J = j) = f_j(x) \tag{2}$$

for each j which is named the conditional density function. These parameters can be used for enumerating posterior probability according to Bayes formulae:

$$p(j|x) = \frac{p_j f_j(x)}{\sum_{j=1}^{M} p_j f_j(x)}. \tag{3}$$

The formalisation of the recognition task leads to the setting of the optimal Bayes decision algorithm $\Psi(x)$, which minimizes the expected value of the so-called loss function which describes the cost of wrong classification [12]. For the well known 0-1 loss function the mentioned classifier assures the lowest value of the probability of misclassification, and the decision rule chooses the class for which the posterior probability achieved the highest values.

The hierarchical classifier contains a sequence of actions [13-14], see Fig 1. These actions are simple classification tasks executed in the individual nodes of the decision tree. Some specific features are measured on every level of the decision tree. At the first stage features x_0, at the second features x_1 are measured, and so on. Every set of features comes from the whole vector of features. In every node of the decision tree

the classification is executed according to the specific rule. The decisions $i_1, i_2, ..., i_N$ are the results of recognition in the suitable node of the tree. At the last N-th stage, the decision made i_N indicates a single class. This class is the result of the hierarchical classifier.

In our task of classification the number of classes is equal NC. The logic of making the decision is represented using the decision tree. The design of a decision tree structure in our approach to the hierarchical classifier is based on human expert knowledge. The terminal nodes are labeled with the number of the classes from the M={1,2,...,NC}, where M is the set of labels classes. The non-terminal are labeled by numbers of 0, NC+1, NC+2… reserving 0 for the root-node.

Let us introduce the notation for the received model of multistage recognition [9]:

$M(n)$ – the set of nodes, which distance from the root is n, $n = 0,1,2,...,N$. In particular $M(0) = \{0\}$, $M(N) = M$,

$\overline{M} = \bigcup_{n=0}^{N-1} M(n)$ – the set of interior nodes (non terminal),

$M_i \subseteq M(N)$ – the set of class labels attainable from i-th node ($i \in \overline{M}$),

M^i – the set of nodes of the immediate descendant node i ($i \in \overline{M}$),

m_i – the node of the direct predecessor of i-th node ($i \neq 0$),

$s(i)$ – the set of nodes on the path from the root-node to i-th node, $i \neq 0$..

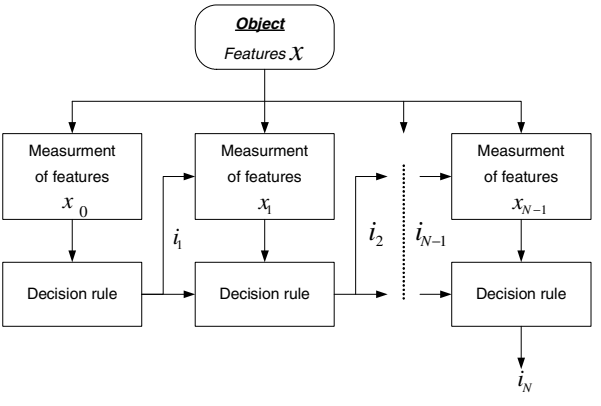


Fig. 1. Block diagram of the hierarchical classifier

Our target now is to calculate the so-called multistage recognition strategy $\pi_N = \{\psi_i\}_{i \in \bar{M}}$, that is the set of recognition algorithms in the form [12]:

$$\Psi_i : X_i \to M^i, \quad i \in \bar{M}. \tag{4}$$

The presented formula is the decision rule (recognition algorithm) used at i-th node which maps observation subspace to the set of immediate descendant nodes of i-th node.

The strategy of the decision tree classifier represents the logic of making the decision. We favor two cases of the decision strategy. The first one is the locally optimal strategy. This strategy consists in minimizing the misclassification rate for the particular nodes of a tree. Its decision rules are mutually independent. There are no relationships between the nodes. The recognition algorithm at the *n*-th stage is as follows:

$$\overline{\Psi}_{i_n}(x_{i_n}) = i_{n+1} \text{ when } \arg\max_{l \in \mathbb{M}^{i_n}} p(k) f_l(x_{i_n}) . \tag{5}$$

The second is globally optimal strategy. This strategy minimizes the mean probability of misclassification. The decision rules are mutually dependent by the empirical probability of the correct classification. The recognition algorithm at the *n*-th stage is as follows:

$$\Psi^*_{i_n}(x_{i_n}) = i_{n+1} \text{ when } \arg\max_{l \in \mathbb{M}^{i_n}} Pc(l) p(l) f_l(x_{i_n}) , \tag{6}$$

where $Pc(l)$ is the empirical probability of the correct classification at the next stages if at the *n*-th stage decision i_{n+1} is made.

3 Experiments

The first mathematical model of acute abdominal pain (APP) with the decision tree was given in [8]. This model has sixteen classes and four stages of recognition. We simplified it to eight classes and two stages (see Fig.2).

It leads to the following classification of the AAP:

1. cholecystitis,
2. pancreatitis,
3. non-specific abdominal pain,
4. rare disorders of “acute abdominal”,
5. appendicitis,
6. divercitulitis,
7. small-bowel obstruction,
8. perforated peptic ulcer.

The expert physicians (from the Surgical Clinic Wroclaw Medical Academy) gave the decision tree presented in Fig.2. Numbers of leafs are the number of diagnosis and the numbers in the nodes correspond to the following diagnoses:

9. acute enteropathy,
10. acute disorders of the digestive system,
11. others.

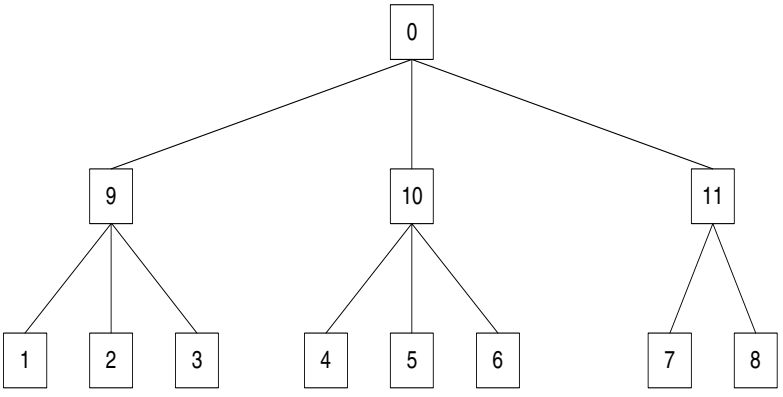


Fig. 2. Block diagram of the hierarchical classifier

4 Experimental Investigation

In the experiment we used the multistage classifiers with the heuristic decision tree. In the interior node the different classifiers were used. The locally optimal strategy was considered. The dataset used in the experiment was collected in Surgical Clinic, Wroclaw Medical Academy and consists of 476 clinical histories. The clinical feature description is presented in Tab.1. Features used in each node of the decision tree are presented in Tab. 2. The selection of features has been made in accordance with the suggestions of another work on this topic [5], [9].

Diagnostic accuracy (DA) was defined by the number of correct predictions divided by the total number of elements in the dataset. The aim of the experiment is to compare the errors for different classifiers and their parameters.

Table 1. Clinical feature description

no	Attribute	no	Attribute	no	Attribute
1	Sex	12	nausea and vomiting	23	Pulse
2	Age	13	Appetite	24	respiratory movements of abdomen
3	pain location on the beginning	14	bowel movement	25	Flatulence
4	pain location on present	15	Urinate	26	Tenderness (location)
5	pain intensity	16	previous indigestion	27	Blumberg's sign

Table 1. (*continued*)

6	aggravating factors	17	Jaundice	28	muscle's Demence
7	relieving factors	18	previous surgery (abdominal)	29	increased tension of abdominal
8	pain progression	19	Drugs	30	Swellings
9	pain duration	20	Mood	31	Murphy's sign
10	pain type on the beginning	21	skin's color		
11	pain type at present	22	Temperature		

Table 2. Features used in nodes of decision tree

node	Features						
0	26	30	11	6	16	31	10
9	26	3	19	4	31	16	10
10	27	3	11	6	28	15	26
11	27	11	29	6	26	10	23

The values of error rate for one stage classification are presented in Tab. 3. Tab.4-7 show the values of error rate on the first and second stage of the classification respectively.

Table 3. Error rate for one stage classification for the set of features {26, 30, 11, 6, 16, 31, 10}

	Method of classification						
	k-NN		Decision tree		Neutral networks		Ensemble
	k=5	k=9	ProbF	Variance	ne=2	ne=3	
Misclass. rate	**0,328**	0,419	0,377	0,377	0,405	0,419	0,377

Table 4. Error rate on the first stage of classification – node "0"

	Method of classification						
	k-NN		Decision tree		Neutral networks		Ensemble
	k=5	k=9	ProbF	Variance	ne=2	ne=3	
Misclass. rate	0,326	0,333	0,326	0,284	0,333	0,347	**0,263**

Table 5. Error rate on the second stage of classification – node "9"

	Method of classification						
	k-NN		Decision tree		Neutral networks		Ensemble
	k=5	k=9	ProbF	Variance	ne=2	ne=3	
Misclass. rate	0,304	0,260	**0,043**	**0,043**	0,130	0,108	0,086

Table 6. Error rate on the second stage of classification – node "10"

	Method of classification						
	k-NN		Decision tree		Neutral networks		Ensemble
	k=5	k=9	ProbF	Variance	ne=2	ne=3	
Misclass. rate	0,355	0,555	0,222	0,222	**0,166**	**0,166**	0,222

Table 7. Error rate on the second stage of classification – node "11"

	Method of classification						
	k-NN		Decision tree		Neutral networks		Ensemble
	k=5	k=9	ProbF	Variance	ne=2	ne=3	
Misclass. rate	0,0	0,0	0,0	0,0	0,0	0,0	0,0

In our experiment, we used several concepts of classifiers. In the study k-NN, the decision tree, neutral networks and the ensemble classification model were used. K-NN classifier was tested for k=5 and k=9. In the decision tree we used two splitting rules (ProbF — p-value of F-test associated with node variance, variance — reduction in the square error from node means). In the neutral networks we used multilayer perceptron which has one hidden layer, with a linear combination functions in the hidden and output layers and a sigmoid activation functions in the hidden layers. The experiments were carried out for the number of neurons (ne) equal to two or three. An ensemble classifier with majority voting was also used. In this model, all of the classifiers were in the ensemble. The experiments were carried out in SAS Enterprise Miner 6.1 environment [15].

5 Discussion

The following conclusions could be drawn from the experiment: At node 11 we deal with the error-free classification. In different multi-stage classifier nodes different classification methods were the best. In node "0" the ensemble classifier method was the best. In node "9" and "10" the decision tree and neutral networks were the best respectively. For one stage classification k-NN rule with k=5 on the other hand was the best.

In the previous study only k-NN rule was applied. The results presented in this work suggest that you can still improve the quality of classification for this problem of medical diagnosis. In particular, further research may relate to changes in the structure of the multistage classifier and the use of the preprocessing method for selection of the features. In this work we used the previously proposed decision tree structure and selection of features in each node.

6 Conclusion

The recognition methods based on the hierarchical approach was presented. The different classifier methods were applied to the medical decision problem (recognition of Acute Abdominal Pain) and can be used to help the clinicians to make their own diagnosis.

The presented heuristic results for the choice of the classification method in the interior node of the decision tree demonstrate the effectiveness of the proposed concepts in such computer-aided medical diagnosis problems. Further research may relate to changes in the structure of the decision tree and use another multistage recognition strategy.

Acknowledgements. This work is supported in part by the National Science Centre under the grant which is being realized in years 2011-2014.

References

1. Mui, J., Fu, K.S.: Automated classification of nucleated blood cells using a binary tree classifier. IEEE Trans. Pattern Anal. PAMI-2, 429–443 (1980)
2. Wozniak, M.: Two-Stage Classifier for Diagnosis of Hypertension Type. In: Maglaveras, N., Chouvarda, I., Koutkias, V., Brause, R. (eds.) ISBMDA 2006. LNCS (LNBI), vol. 4345, pp. 433–440. Springer, Heidelberg (2006)
3. Penar, W., Wozniak, M.: Cost sensitive methods of constructing hierarchical classifiers. Expert Systems 27(3), 146–155 (2010)
4. Kołakowska, A., Malina, W.: Fisher Sequential Classifiers. IEEE Transactions on Systems, Man and Cybernetics, Part B 35(5), 988–998 (2005)
5. Kurzyński, M.: On the Multistage Bayes Classifier. Pattern Recognition 21, 355–365 (1988)

6. De Dombal, F.T., Leaper, D.J., Staniland, J.R., McCann, A.P., Horrocks, C.: Computer-aided diagnosis of acute abdominal pain. Br. Med. J. II, 9–13 (1972)
7. Eich, H.P., Ohmann, C., Lang, K.: Decision support in acute abdominal pain using an expert system for different knowledge bases. In: Proceedings of the 10th IEEE Symposium on Computer-Based Medical Systems, pp. 2–7 (1997)
8. Kurzyński, M.: Diagnosis of acute abdominal pain using three-stage classifier. Computers in Biology and Medicine 17(1), 19–27 (1987)
9. Burduk, R., Woźniak, M.: Bayes Multistage Classifier and Boosted C4.5 Algorithm in Acute Abdominal Pain Diagnosis. In: Cyran, K.A., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. AISC, vol. 59, pp. 371–378. Springer, Heidelberg (2009)
10. Ohmann, C., Moustakis, V., Yang, Q., Lang, K.: Evaluation of automatic knowledge acquisition techniques in the diagnosis of acute abdominal pain. Artif. Intell. Med. 8(1), 23–36 (1996)
11. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice Hall, London (1982)
12. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley and Sons (2000)
13. Burduk, R., Kurzyński, M.: Two-stage binary classifier with fuzzy-valued loss function. Pattern Analysis and Applications 9(4), 353–358 (2006)
14. Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. IEEE Trans. Systems, Man Cyber. 21(3), 660–674 (1991)
15. Getting Started with SAS Enterprise Miner 6.1, http://support.sas.com/documentation/onlinedoc/miner

Ensemble of Tensor Classifiers Based on the Higher-Order Singular Value Decomposition

Bogusław Cyganek

AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@agh.edu.pl

Abstract. In this paper we present an ensemble composed of classifiers operating with multi-dimensional data. Classification is performed in tensor spaces spanned by the basis obtained from the Higher-Order Singular Value Decomposition of the pattern tensors. These showed superior results when processing multi-dimensional data, such as sequences of images. However, multi-dimensionality leads to excessive computational requirements. The proposed method alleviates this problem, first by partitioning the input dataset, and then by feeding each partition into a separate tensor classifiers of the ensemble. Despite the computational advantages, also accuracy of the ensemble showed to be higher compared to a single classifier case. The method was tested in the context of object recognition in computer vision. In the paper we discuss also methods of input image prefiltering in order to increase accuracy. The conducted experiments show high efficacy of the proposed solution.

Keywords: Classification, HOSVD, ensemble of classifiers.

1 Introduction

Object recognition by computers requires construction of accurate and fast classifiers. There are many examples of these which are reported in literature [16][18][19]. However, many existing methods do not account for the multi-dimensionality of the classified data. Recently, this issue was addressed with help of the tensor analysis. One of the first methods which utilize this approach was face recognition system proposed by Vasilescu *et al.* [19]. In their approach tensors constitute a major mathematical tool to cope with multiple factors of face patters, which can be represented under different poses, views, illuminations, etc. Because of this, the method was called tensor-faces. Another system that is based on tensor analysis was proposed by Savas *et al.* [15]. It was applied to the problem of handwritten digits recognition [13]. The method by Savas *et al.* assumes tensor decomposition which allows an equivalent representation of a tensor as a product of a core tensor and the unitary mode matrices. This decomposition is known as the Higher-Order Singular Value Decomposition (HOSVD), and is related to the Tucker decomposition [1][12][9]. A version of this method was then used by Cyganek in the system for road signs recognition [3]. In this case the input tensor is built from artificially generated

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 578–589, 2012.

deformed versions of the prototype road sign exemplars. All these systems, which are based on HOSVD, show very high accuracy and high speed of response. However, computation of the HOSVD from large size tensors is computationally demanding. The decomposition algorithm requires computation of a sequence of SVD decompositions of matrices obtained from the flattened input tensor. However, dimensions of these matrices can be huge since they are multiplications of all dimensions of the input tensor. In many applications this can be very problematic. The method presented in this paper shows how to alleviate this problem by construction of an ensemble of tensor-based classifiers. In the proposed solution tensors are of much smaller size than in a case of a single tensor based classifier. As it will be shown, despite the computational advantages, the proposed ensemble shows also superior accuracy when compared with a single classifier.

The method was tested in the task of handwritten digits recognition and showed good results and high speed of operation. Detailed experimental results are provided for the highly demanding USPS dataset [6][20]. These verified our assumption on high accuracy and lower computational complexity of the proposed ensemble.

The rest of the paper is organized as follows: Section 2 presents basics on N-Mode Principal Component Analysis for pattern recognition. In Section 3 we provide details of the proposed methods, i.e. construction of the ensemble of HOSVD classifiers. Experimental results with discussion of the obtained scores are provided in Section 4. The paper ends with conclusions in Section 5.

2 N-Mode Principal Component Analysis for Pattern Recognition

Tensors in data mining and classification are defined as multidimensional arrays. They generalize such concepts as scalars, vectors, and matrices, which all are tensors. Processing and analysis of images builds well into this framework due to a multi-dimensional nature of data in video streams. However, an analysis of contents of a video represented as tensors requires their proper decomposition. There are many tensor decompositions which allow either their analysis or compact representation. However, one of the most popular is the already mentioned HOSVD [1][11][9]. As it will be shown in the next sections, HOSVD can be used to build orthogonal spaces which can be then used for pattern recognition in a similar way to the standard PCA based classifiers [4][18]. However, before we show properties and the algorithm of computation of the HOSVD, we present briefly some of the most important concepts of the tensor algebra.

2.1 Tensor Algebra Concepts

Although tensors can be multi-dimensional, in many methods it is convenient to represent them in the matrix-like, or flattened, form. More specifically, for a *P-th* order tensor $\mathcal{T} \in \Re^{N_1 \times N_2 \times \ldots N_P}$, the *k-mode vector* of $\mathcal{T}$ is defined as a vector obtained from the elements of $\mathcal{T}$ by varying only one index n_k, while keeping all other fixed. Let us assume that from the tensor $\mathcal{T}$ the following matrix

$$\mathbf{T}_{(k)} \in \Re^{N_k \times \left(N_1 N_2 \ldots N_{k-1} N_{k+1} \ldots N_P\right)} \tag{1}$$

is formed. Now columns of $\mathbf{T}_{(k)}$ are *k-mode* vectors of $\mathcal{T}$. The *k*-mode representation of a tensor is obtained by selecting the *k*-th index which becomes a row index of its flatten representation. On the other hand its column index is a product of all other *P*-1 indices. Nevertheless, where an element of the tensor is stored in memory depends on the chosen permutation of these *P*-1 indices, which results in (*P*-1)! possibilities. From these only two, i.e. forward and backward cycle modes, are used [11].

The second key concept is a *k*-mode multiplication of a tensor $\mathcal{T} \in \Re^{N_1 \times N_2 \times \ldots N_P}$ and a matrix $\mathbf{M} \in \Re^{Q \times N_k}$. In result of such a multiplication a tensor $\mathcal{S} \in \Re^{N_1 \times N_2 \times \ldots N_{k-1} \times Q \times N_{k+1} \times \ldots N_P}$ is obtained, which elements can be expressed as follows

$$\mathcal{S}_{n_1 n_2 \ldots n_{k-1} q n_{k+1} \ldots n_P} = \left(\mathcal{T} \times_k \mathbf{M}\right)_{n_1 n_2 \ldots n_{k-1} q n_{k+1} \ldots n_P} = \sum_{n_k=1}^{N_k} t_{n_1 n_2 \ldots n_{k-1} n_k n_{k+1} \ldots n_P} m_{q n_k}. \tag{2}$$

Finally, to analyze contents of a tensor a proper decomposition needs to be applied. The HOSVD decomposition, used also in the proposed method, allows any *P*-dimensional tensor $\mathcal{T} \in \Re^{N_1 \times N_2 \times \ldots N_m \times \ldots N_n \times \ldots N_P}$ to be equivalently represented in the following form [11][12]

$$\mathcal{T} = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_P \mathbf{S}_P . \tag{3}$$

In the above, $\mathbf{S}_k$ are unitary matrices of dimensions $N_k \times N_k$, called *mode matrices*. $\mathcal{Z} \in \Re^{N_1 \times N_2 \times \ldots N_m \times \ldots N_n \times \ldots N_P}$ is *a core tensor* which fulfills the following properties [11][12]:

1. Two subtensors $\mathcal{Z}_{n_k=a}$ and $\mathcal{Z}_{n_k=b}$, are orthogonal for all possible values of *k* for which $a \neq b$, i.e.

$$\mathcal{Z}_{n_k=a} \cdot \mathcal{Z}_{n_k=b} = 0 , \tag{4}$$

2. All subtensors of $\mathcal{Z}$ for all *k* can be ordered according to their Frobenius norms

$$\left\| \mathcal{Z}_{n_k=1} \right\| \geq \left\| \mathcal{Z}_{n_k=2} \right\| \geq \ldots \geq \left\| \mathcal{Z}_{n_k=N_P} \right\| \geq 0 , \tag{5}$$

Finally, the *a-mode* singular value of $\mathcal{T}$ is defined as follows

$$\left\| \mathcal{Z}_{n_k=a} \right\| = \sigma_a^k . \tag{6}$$

In the next section we discuss an algorithm for computation of the HOSVD.

2.2 Computation of the Higher-Order Singular Value Decomposition

Lathauwer proposed a method of computation of the HOSVD which is based on successive application of the SVD decompositions to the flattened matrices of a given tensor [11]. Thus, HOSVD of a *P*-dimensional tensor $\mathcal{T}$ is presented by the following algorithm [11][12]:

1. For each k=1, …, P do:
  a. Flatten tensor $\mathcal{T}$ to obtain $\mathbf{T}_k$, from Eg.(1)
  b. Compute $\mathbf{S}_k$ from the SVD decomposition of $\mathbf{T}_k$

$$\mathbf{T}_k = \mathbf{S}_k \mathbf{V}_k \mathbf{D}_k^T \tag{7}$$

2. Using all $\mathbf{S}_k$ compute the core tensor:

$$\mathcal{Z} = \mathcal{T} \times_1 \mathbf{S}_1^T \times_2 \mathbf{S}_2^T \ldots \times_P \mathbf{S}_P^T \tag{8}$$

Fig. 1. An algorithm for computation of the HOSVD of tensors

Because $\mathbf{S}_k$ are orthogonal, the core tensor $\mathcal{Z}$ can be expressed as

$$\mathbf{Z}_{(k)} = \mathbf{S}_k^T \mathbf{T}_{(k)} \left[\mathbf{S}_{k+1} \otimes \mathbf{S}_{k+2} \otimes \ldots \otimes \mathbf{S}_P \otimes \mathbf{S}_1 \otimes \mathbf{S}_2 \otimes \ldots \otimes \mathbf{S}_{k-1} \right]. \tag{9}$$

From the algorithm in Fig. 1 we see that computation of the HOSVD requires a sequence of SVD decomposition of matrices obtained from the input tensor. However, dimensions of these matrices can be very large since they are just multiplications of all dimensions of the decomposed tensor, as shown in Eg. (1). In many practical cases this poses a real problem. The method presented in this paper shows how to overcome this problem with construction of an ensemble of tensors, which are of lower sizes, however. As will be shown, apart from this computational advantage, such ensemble of classifiers shows also better accuracy when compared with a single classifier.

2.3 Pattern Recognition in the Tensor Spanned Spaces

For each mode matrix $\mathbf{S}_i$ in (3) the following sum can be constructed

$$\mathcal{T} = \sum_{h=1}^{N_P} \mathcal{T}_h \times_P \mathbf{s}_P^h , \tag{10}$$

thanks to the commutative properties of the *k*-mode multiplication. In the above

$$\mathcal{T}_h = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_{P-1} \mathbf{S}_{P-1} \tag{11}$$

constitute the basis tensors and $\mathbf{s}^h_P$ are columns of the unitary matrix $\mathbf{S}_P$. Because $\mathcal{T}_h$ is of dimension P-1 then $\times_P$ in (10) is an outer product, i.e. a product of two tensors of dimensions P-1 and 1. Moreover, due to the orthogonality properties of the core tensor $\mathcal{Z}$ in (11), $\mathcal{T}_h$ are also orthogonal. Thus, they can constitute a basis which spans a sub-space. This property is used to construct a HOSVD based classifier, as follows.

In the tensor space spanned by $\mathcal{T}_h$, pattern recognition can be stated as testing a distance of a given test pattern $\mathbf{P}_x$ to its projections in each of the spaces spanned by the set of the bases $\mathcal{T}_h$ in (11). This can be expressed as the following minimization problem [15]

$$\min_{i,c^i_h} \underbrace{\left\| \mathbf{P}_x - \sum_{h=1}^{H} c^i_h \mathcal{T}^i_h \right\|^2}_{Q_i}, \tag{12}$$

where the scalars c^i_h denote unknown coordinates of $\mathbf{P}_x$ in the space spanned by $\mathcal{T}^i_h$, and $H \leq N_P$ denotes a number of chosen dominating components.

To solve (12) the squared norm Q of (12) is created for a selected i. Assuming further that $\mathcal{T}^i_h$ and $\mathbf{P}_x$ are normalized the following is obtained (the *hat* indicates normalized tensors)

$$\rho_i = 1 - \sum_{h=1}^{H} \left\langle \hat{\mathcal{T}}^i_h, \hat{\mathbf{P}}_x \right\rangle^2. \tag{13}$$

Thus, to minimize (12) we need to maximize the following value

$$\hat{\rho}_i = \sum_{h=1}^{H} \left\langle \hat{\mathcal{T}}^i_h, \hat{P}_x \right\rangle^2, \tag{14}$$

In other words, the HOSVD based classifier returns a class i for which its ρ_i from (14) is the largest.

3 Construction of the Ensemble of HOSVD Classifiers

It was shown that an ensemble of even simple classifiers can perform better that a complex but single classifier [10][14][7]. As we will show, this holds also for the HOSVD based classifiers. Apart from the higher overall accuracy, an ensemble of HOSVD classifiers offers also computational advantages such as lower memory requirements due to reduced training data partitions, as well as the parallel run-time structure.

To build an ensemble of HOSVD based classifiers we propose to use bagging and image preprocessing. Bagging consists in creating a number of variants $\mathbf{X}_T^{(i)}$ of the training set $\mathbf{X}_T$, by a uniform data sampling from $\mathbf{X}_T$ with replacement (a bootstrap

aggregation method) [17]. It was shown that bagging can reduce variance of a classifier and improve its generalization properties [5]. Each data variant is used to train a separate member of the ensemble, which in our case is the HOSVD classifier. A number of data in each variant $\mathbf{X}_T^{(i)}$ is one of the training parameters of the ensemble. However, the issue is that in the case of an ensemble of classifiers, each classifier can be trained with data variant which is less numerous than the maximally available number of training points and the overall accuracy of the ensemble will be still higher that for a single classifier case. Thanks to this strategy we can cope easily with massive data, training one classifier at a time. Additionally, it is possible to extend the ensemble with new members if new training data are obtained in the later time. An analysis of the experimental results confirms these suppositions.

Additionally, to improve performance of the classifiers we propose to add image prefiltering stage, as shown in Fig. 2. In our experiments we tested the following prefilters:

1. Affine image warping (bilinear interpolation);
2. Edge detection with the Savitzky-Golay filters;
3. Phase component of the structural tensor;
4. Census transform;
5. Gaussian noise addition.

Detailed algorithms with code examples of the above procedures are described in [2].

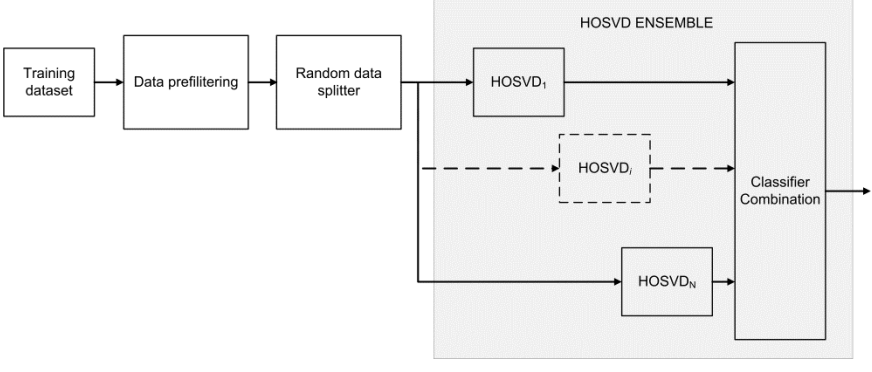


Fig. 2. Data flow in the proposed construction of the HOSVD ensemble of classifiers

Fig. 2 shows the construction steps (white blocks), as well as structure of the HOSVD ensemble (gray blocks). Each of the individual HOSVD$_i$ experts in Fig. 2 is trained to recognize exactly ten classes of the handwritten digits, however each is fed with different dataset. That is, each HOSVD$_i$ is trained with its specific data partition obtained by random sampling a number of training data (i.e. images of the handwritten digits) from the set of all data (bagging). This is done in the random splitter block in Fig. 2, using a random generator with a uniform distribution.

Summarizing, the training parameters of the entire ensemble are as follows:

1. Number of member classifiers in the ensemble;
2. Size of data partitions in the bagging process;
3. Type of the preprocessing filter;
4. Number of components H considered in (14);

In the run-time, answers of each of the classifiers are combined with the majority voting scheme [8][10]. Summarizing, the proposed ensemble of HOSVD classifiers offers the following advantages:

- Significant reduction of memory requirements during training;
- Reduction of computations (due to lower size of the matrices);
- Possible incremental build (e.g. if new training samples are coming at later time, they can compose a new member of the ensemble);
- Proper data prefiltering can result in up to 1-2% improvement (however, for different datasets, this needs experimental verification);
- The method naturally leads to the parallel training and run-time architectures;

Certainly, the most important factor is the overall accuracy of an ensemble. As we show in the next section, this and also all of the aforementioned postulates were verified experimentally.

4 Experimental Results

The presented method was entirely implemented in C++, supported by the HIL library [2]. The experiments were run on the computer with 8 GB RAM and with Pentium® Quad Core Q 820 (clock 1.73 GHz). For the experiments the USPS dataset was used [6][20]. The same set was also used by Savas *et al.* [15]. This dataset contains selected and preprocessed scans of the handwritten digits from envelopes by the U.S. Postal Service. Fig. 3 depicts exemplars of each digit from the training set (Fig. 3a), and from the testing set (Fig. 3b), respectively.

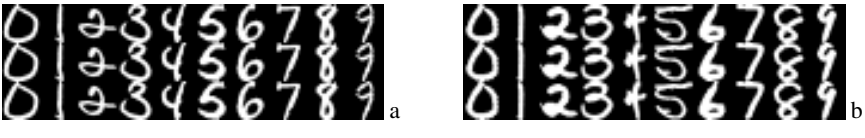


Fig. 3. Visualization of the two data sets from the ZIP database (from data). Ten exemplars of each digit from the training set (a), and from the testing set (b).

Each test and train pattern originally is in a form of a 16×16 gray level image. Since this dataset is perceived as a relatively difficult for machine classification (reported human error is 2.5%), it has been used for comparison of different classifiers [13][15]. The database is divided into the training and testing partitions,

counting 7291 and 2007 exemplars, respectively. Detailed numbers of training and test patterns for each digit are contained in Table 1.s

Each experimental setup was run number of times (from 3 to 10, depending on computational complexity) and an average answer is reported. In all cases the Gaussian noise was added to the input image at level of 10%, in accordance with the procedure described in [2].

In the first group of experiments we tested influence of the number of data points used in bagging process on accuracy of the ensembles with different number of members. Fig. 4 shows obtained results of these tests in respect to the number of classifiers in the ensembles. We see that the best accuracies were obtained for partitions of 560 points. However, almost similar accuracies are for partitions counting 256 points. It is interesting to observe differences of accuracy for a single classifier system (i.e. one classifier) and ensembles even with only 2 or 3 members. In this case accuracy increases rapidly by about 1%, regardless of a number of data points used in bagging. This acknowledges our assumption of better accuracy in the case of the ensembles with diversity obtained with data bagging. At the same time, each HOSVD classifier of an ensemble requires much less memory during training since it deals with a smaller training set.

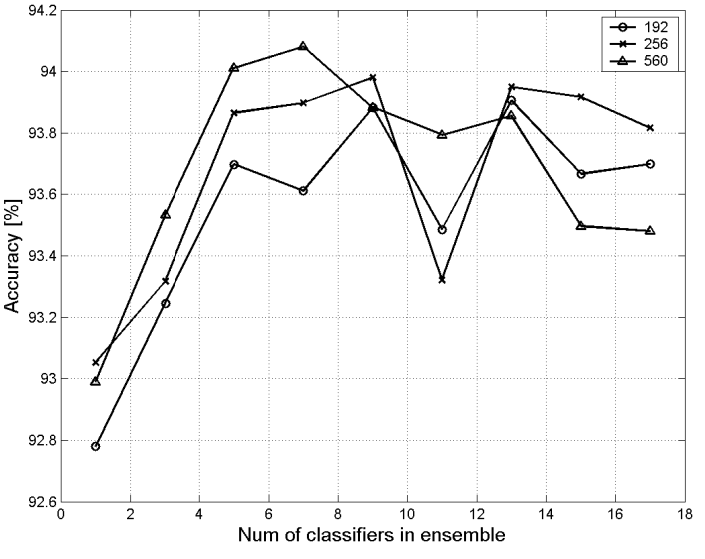


Fig. 4. Accuracy vs. number of classifiers in the ensemble for three sizes of data samples in the bagging process (192, 256, 560). Input images of size 16×16. In Eq. (14) number of components H=16.

In the next experiments we compared performance of the ensembles of different number of members in respect to the two different sizes of the input images: 16×16 vs. 32×32 pixels. Fig. 5 depicts obtained accuracies in these experiments. It is evident that images of larger size always result in higher accuracy. Also a difference between

a single classifier and an ensemble of two or more members is not so rapid in the case of larger input images. The other tests with other of the aforementioned prefilters which changed signal representation such as, computing edges, phases of the structural tensor, as well as computing the Census measure did not produce better accuracies than bare intensity signal. However, as it was already pointed out, the interpolation to larger size increases accuracy. Unfortunately, this is burdened with higher computational costs. Thus, in our setup a trade-off was set to the resolution of 32×32 pixels. Larger images did not produce much better accuracy, whereas computational demands grew excessively.

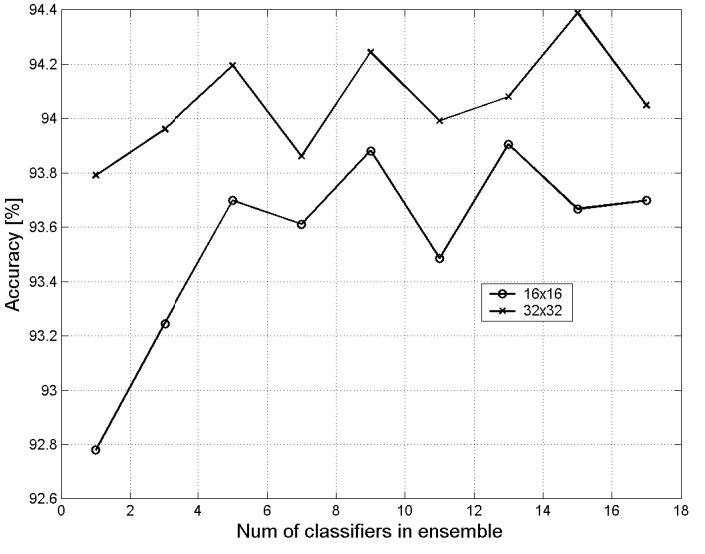


Fig. 5. Accuracy vs. number of classifiers in the ensemble for two different sizes of input images: original (16×16) vs. enlarged by image warping (32×32). In both cases 192 data samples (images) were used in bagging. In Eq. (14) number of used components H=16.

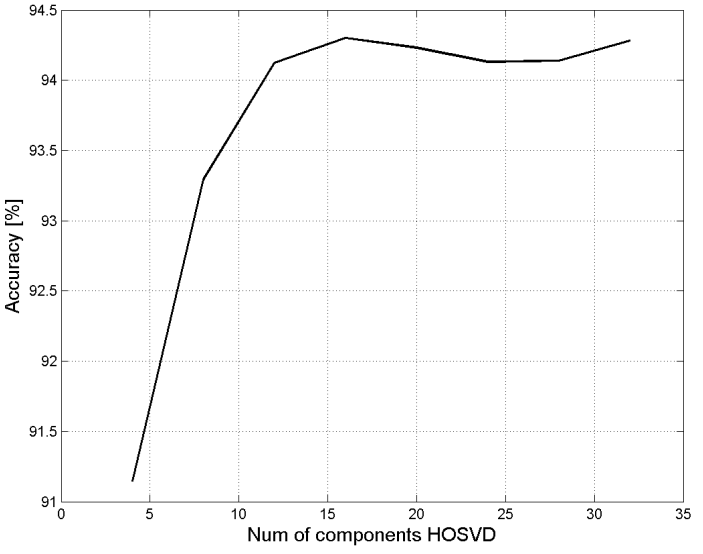


Fig. 6. Accuracy vs. number of components H in Eq. (14). Input images were warped to 32×32, number of classifiers in ensemble set to 15, bagging partitions of 192 images used.

In the next experiment we tried to figure out the optimal value of the number of components H of the HOSVD approximation in Eq. (14). Fig. 6 depicts accuracy vs. parameter H. In this case a setup with best accuracy was chosen (see Fig. 5). That is, the input images were warped to 32×32 pixels, number of classifiers in the ensemble was set to 15, bagging partitions of 192 images. It is evident that there is an optimal value of H=16. Beyond that point, when H is increased, accuracy reaches its plateau. The reason of this is that increasing H we consider components usually associated with noise (like in the PCA for matrices). Further experiments with different settings revealed very similar value of the parameter H. The reported value of H by Savas *et al.* is 12 [15]. However, in our setups we deal with ensembles, so the values of H can differ slightly.

Table 1 shows numbers of training and test patterns in the dataset. It shows also detailed accuracies for each digit separately for the ensemble which reached the best accuracy in Fig. 5. This ensemble counts 15 members, input data were transformed to 32×32 resolution, 192 training images were used, and number of components in (14) is H=16.

Table 1. Accuracies and parameters for each digit separately. Experimental setup: 15 members in the ensemble, input data transformed to 32×32 resolution, 192 training images from bagging, H=16 components.

Digit	0	1	2	3	4	5	6	7	8	9
#Train	1194	1005	731	658	652	556	664	645	542	644
#Test	359	264	198	166	200	160	170	147	166	177
Accur.	0.986	0.981	0.899	0.880	0.930	0.925	0.982	0.959	0.910	0.960

Results in Table 1 show that the worse detailed accuracy was obtained for digit "3". Indeed it can be confused with "8", for which obtained accuracy is also not the best one. The best particular accuracies were obtained for "0", "1", and "6". Measured average classification time for a single data point is in order of 1-2 ms.

5 Conclusions

In this paper we propose a novel method of construction of the ensemble of tensor classifiers based on the Higher-Order Singular Value Decomposition. It was shown that although the HOSVD based classifier allows high accuracy and speed of classification, its construction for large training sets can be computationally demanding. The proposed method alleviates this problem by construction of the ensemble of the HOSVD classifiers and data bagging technique. Thanks to this, the tensors for the HOSVD decompositions are of lower size than the original tensor which would contain all training data. Moreover, the constructed ensemble of classifiers achieves higher overall accuracy as compared to a solution with the single classifier. The method was tested in the context of handwritten digits recognition. In the paper we discussed also methods of prefiltering of the input patterns in order to increase accuracy of the system. In this respect we checked affine warping of images, addition of noise, Census transformation, as well as the structural tensor. We showed that application of the first of the mentioned filters can increase accuracy of the system. The obtained results acknowledged our assumptions.

Acknowledgement. The work was supported in the years 2011-2012 from the funds of the Polish National Science Centre NCN, contract no. DEC-2011/01/B/ST6/01994.

References

1. Cichocki, A., Zdunek, R., Amari, S.: Nonnegative Matrix and Tensor Factorization. IEEE Signal Processing Magazine 25(1), 142–145 (2008)
2. Cyganek, B., Siebert, J.P.: An Introduction to 3D Computer Vision Techniques and Algorithms. Wiley (2009)
3. Cyganek, B.: An Analysis of the Road Signs Classification Based on the Higher-Order Singular Value Decomposition of the Deformable Pattern Tensors. In: Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2010, Part II. LNCS, vol. 6475, pp. 191–202. Springer, Heidelberg (2010)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley (2001)
5. Grandvalet, Y.: Bagging equalizes influence. Machine Learning 55, 251–270 (2004)
6. Hull, J.: A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(5), 550–554 (1994)
7. Jackowski, K., Woźniak, M.: Algorithm of designing compound recognition system on the basis of combining classifiers with simultaneous splitting feature space into competence areas. Pattern Analysis and Applications 12, 415–425 (2009)
8. Kittler, J., Hatef, M., Duing, R.P.W., Matas, J.: On Combining Classifiers. IEEE PAMI 20(3), 226–239 (1998)

9. Kolda, T.G., Bader, B.W.: Tensor Decompositions and Applications. SIAM Review, 455–500 (2008)
10. Kuncheva, L.I.: Combining Pattern Classifiers. Methods and Algorithms. Wiley Interscience (2005)
11. de Lathauwer, L.: Signal Processing Based on Multilinear Algebra. PhD dissertation, Katholieke Universiteit Leuven (1997)
12. de Lathauwer, L., Moor de, B., Vandewalle, J.: A Multilinear Singular Value Decomposition. SIAM Journal of Matrix Analysis and Applications 21(4), 1253–1278 (2000)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proc. IEEE on Speech & Image Processing 86(11), 2278–2324 (1998)
14. Polikar, R.: Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine, 21–45 (2006)
15. Savas, B., Eldén, L.: Handwritten digit classification using higher order singular value decomposition. Pattern Recognition 40, 993–1003 (2007)
16. Szeliski, R.: Computer Vision. Algorithms and Applications. Springer, Heidelberg (2011)
17. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Academic Press (2009)
18. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3(1), 71–86 (1991)
19. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear Analysis of Image Ensembles: TensorFaces. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 447–460. Springer, Heidelberg (2002)
20. `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`

Combining Diverse One-Class Classifiers

Bartosz Krawczyk and Michał Woźniak

Department of Systems and Computer Networks,
Wroclaw University of Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{bartosz.krawczyk,michal.wozniak}@pwr.wroc.pl

Abstract. Multiple Classifier Systems (MCSs) are the focus of intense research and a large variety of methods have been developed in order to exploit strengths of individual classifiers. In this paper we address the problem how to implement a multi-class classifier by an ensemble of one-class classifiers. To improve the performance of a compound classifier, different individual classifiers (which may e.g., differ in complexity, type, training algorithm or other) can be combined and that could increase its both performance, and robustness. The model of one-class classifiers is dedicated to recognize one class only, therefore it is a quite difficult to produce MCSs on the basis of it. One of the important problem is how to ensure diversity of classifier ensemble which consists of one-class classifiers. Well-known diversity measures have been developed for committees of multiclass classifiers. In this work we propose a novel diversity measure which can be applied to a set of one-class classifiers. Additionally we propose a classifier fusion model dedicated to one-class classifiers, which allows more than one classifier per class. We will try answer the question if increasing number of individual one-class classifier has an impact on quality of MCS. The proposed model was evaluated by computer experiments and their results prove that proposed model can outperform well known fusion methods.

Keywords: Pattern recognition, machine learning, one-class classification, classifier ensemble, diversity measure.

1 Introduction

Multiple classifier systems (MCSs), known as combined classifiers raise universal interest of machine learning society and are the focus of intense research from a good thirty years. There are several important issues while building MSCs. One of them is how to select classifiers to ensure the high quality of ensemble. Let's notice that combining similar classifiers could not contribute much to the system being constructed, apart from increasing the computational complexity. That is why it is important to select members of a committee with possibly different components. One of current research is trying to answer the question how the diversity could be measured. Proposed methods exploit several measures which,

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 590–601, 2012.

for example, can be used to minimize the possibility of coincidental failure by different classifiers in the ensemble [19].

A strategy for generating the ensemble members must seek to improve its diversity. We could use varying components of the MCS to enforce classifier diversity:

- using different input data e.g., we could use different partitions of data set or generate various data sets [20], because we hope that classifiers trained on different inputs are complementary;
- using classifiers with different outputs i.e., each individual classifier could be trained to solve subset of M class problem (e.g., binary classifier - one class against the rest strategy) and fusion method should recover the whole set of M classes. The well known technique is Error-Correcting Output Codes (ECOC) [5];
- using classifiers with the same input and output, but trained on the basis of different models or their versions.

Another important issue is the choice of a collective decision making method. There are many different voting methods like majority voting [28] and more advanced types based on weighting the importance of decisions coming from particular committee members [20,9]. Treating the process of weight selection as a separate learning process is an alternative method [16].

The second group of collective decision making methods bases on supports given by individual classifiers for each given classes, the main form of which are the posterior probability estimators, associated with probabilistic models of a given pattern recognition task [17]. One also has to mention many other works, that describe analytical properties and experimental results, as [12]. The aggregating methods, which do not require a learning procedure, use simple operators, like average, maximum, minimum, or product, but they are typically subject to very restrictive conditions [8], which severely limits their practical use.

A multi-class classification problem can be decomposed in the finite quantity of two-class classification problems. Thus, connecting binary classifiers should aim to solve multi-class problem by dividing it into dichotomies. The term single-class classification originates from [18], but also outlier detection [14] or novelty detection [2] are used to name this field of study.

In literature there are several examples of construction of the multi-class classifier by combining the outputs of binary classifiers [15]. 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 [6]. The latter allows one to create neat and intuitive multi-class classifier. In this model at least one binary classifier corresponds to each class. The hypothesis that the given features vector belongs to the selected class is tested against it belonging to one of the other classes. Such an approach has a flow in a case of conflicting answers from classifiers which is not quite straightforward. One-against-all method is usually implemented as so called Winner Takes All (WTA). Each classifier is trained on instances of different class which becomes

the first class, all the other classes correspond to the second one. Final result is achieved by the maximum rule on the values of support for every class. Dieterich and Bakiri [5] propose a combination model, which in case of binary classifier ensembles appeared to be a good extension of approaches mentioned above. Each sequence of bits produced by a set of binary classifiers is associated with codewords during learning. The ECOC method selects a class with the smallest Hamming distance to its codeword. Passerini *et al.* [21] used successfully this scheme for support vector machines. Wilk and Woźniak [27] propose fuzzy combiners of OCCs using ANFIS and develop fuzzy versions of ECOC or decision templates. Another usage of fuzzy logic in related area can be found in [4,3], where the authors propose how to use the fuzzy loss function for designing a compound pattern recognition system.

On the other hand, the combination of one class classifiers still awaits of proper attention [11]. One-class classification (OCC) problem, also called data description, is a special case of binary classification [25]. Their main goal is to detect anomaly or a state other than the one for the target class [26]. It is assumed that only information of the target class is available. The task is to define a boundary around the target class, such that it accepts as much of the target objects as possible, while it minimizes the chance of accepting outlier objects.

The purpose of our contribution focuses on two main actual problems of combining one-class classifiers. We present a novel model of classifier fusion where more than one classifier can be dedicated to each class. Additionally we propose a new diversity measure strictly developed for a pool of a such specific classifiers.

2 Model of Pattern Recognition Task

The aim of pattern recognition is to assign a given object to one of pre-defined categories, on the basis of supplied features. Although it is important for the performance of a classifier, we do not focus on feature selection in this paper, but assume that the set of features is given by an expert or chosen by a feature selection method [7].

A pattern recognition algorithm Ψ maps the feature space X to the set of class labels $\boldsymbol{M}$

$$\Psi : X \to \boldsymbol{M}. \tag{1}$$

This is established on the basis of examples from a training set or rules given by experts. The training set consists of learning examples i.e., observations of features together with their correct classifications.

2.1 One-Class Classification

OCC seeks to distinguish one specific class from the more broad set of classes (e.g., selecting carrot from vegetables, recognizing obstructive nephropathy from various kinds of kidney disorders or identifying medicine-related pictures from an extensive image database). The target class is considered as a positive one, while all other are considered as negative ones. OCC is known as learning in the

absence of examples, as primary object of OCC is to train a classifier using only patterns drawn from the target class distribution. Various terms have been used in the literature to refer to one-class learning approaches.

OCC problems are common in real world where positive examples are widely available but negative ones are hard, expensive or even impossible to gather. For example in bioinformatics it is simple to acquire interesting sequences, but number of possible random mutations is too high to collect all samples of it. In a medical domain positive examples are easy to access (e.g., children with healthy kidneys) and unlabeled data are abundant (e.g., all young patients). At the same time gathering the negative examples (all kinds of kidney disorders in children) is expensive, time-consuming and endangers the health of patients (as most of such test are invasive). Finally all patients in the database cannot be assumed to be negative examples if they have never been tested. So while designing a classification system we have to deal with limited availability of the data. Further applications can be also found in pattern recognition, image retrieval, classification for data mining, rare/unbalanced class classification, etc.

2.2 One-Class Support Vector Machine

One-class SVM classifier (OCSVM) [24] is designed for learning in the absence of examples. This means that it can deal with datasets containing only patterns from one target class. OCSVM classification aims at discriminating one class of target samples from all other ones. It consists of learning the minimum volume contour that encloses most of the data in a given dataset. Its original application is the outlier detection finding data that differ from most of the data within a dataset.

Let $\chi = \{x_1, x_2, ..., x_m\}$ be a given dataset in $\mathbb{R}^d$. Each x_j is a feature vector describing an object. OCSVM use the training data to learn a function $f_\chi : \mathbb{R}^d \mapsto \mathbb{R}$ such that most of the data in χ belong to the set $\mathcal{R}_\chi = \{x \in \mathbb{R}^d; f_\chi(x \geqslant 0)\}$ while the volume of $\mathcal{R}_\chi$ is minimal. This problem is known as $MinimalVolumeSet$ (MVS) estimation. Membership of x to $\mathcal{R}_\chi$ indicates whether this estimated volume is overall similar to χ or not. Therefore when considering a M-class recognition problem we have to learn M membership functions f_{χ_i} - one for each of the classes.

OCSVM uses the following approach to estimate the MVS. A kernel function $k(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$. In the presented paper a Gaussian Radial Basis Function(RBF) kernel is used, such that

$$k(x, x') = exp[- \parallel x - x' \parallel^2 /2\sigma^2], \tag{2}$$

where $\| \cdot \|$ denotes the Euclidean norm in $\mathbb{R}^d$.The kernel induces a new, artificial feature space $\mathcal{H}$ by the usage of mapping $\phi : \mathbb{R}^d \mapsto \mathcal{H}$ dened by $\phi(x) \triangleq k(x, \cdot)$. It has been shown that $\mathcal{H}$ reproduces kernel Hilbert spaces of given functions, with dot product denoted as $\langle \cdot, \cdot \rangle_\mathcal{H}$. The reproducing kernel property implies that

$$\langle \phi(x), \phi(x') \rangle_\mathcal{H} = \langle k(x, \cdot), k(x', \cdot) \rangle_\mathcal{H} = k(x, x'), \tag{3}$$

which makes the evaluation of $k(x, x')$ a linear operation in $\mathcal{H}$, while it is a nonlinear operation in $\mathbb{R}^d$.

Considering the RBF used in this paper it is true that

$$\| \phi(x) \|_{\mathcal{H}}^2 \triangleq \langle \phi(x), \phi(x) \rangle_{\mathcal{H}} = k(x, x) = 1. \tag{4}$$

From this one may assume that all the data mapped into $\mathcal{H}$ are located on the hypersphere with radius equal to one, centered onto the origin of $\mathcal{H}$, which is denoted $S_{(o,R=1)}$. The OCSVM determines in $\mathcal{H}$ the hyperplane $\mathcal{W}$ that separates most of the data from the $S_{(o,R=1)}$, while at the same time maximizing the distance from it. This practically implements the solution to the MVS estimation problem.

Let

$$\mathcal{W} = \{h(\cdot) \in \mathcal{H}; \langle h(\cdot), w(\cdot) \rangle_{\chi} - \rho = 0\}, \tag{5}$$

where parameters $w(\cdot)$ and ρ are the results of the following optimization problem

$$\min_{w,\xi,\rho} \frac{1}{2} \| w(\cdot) \|_{\mathcal{H}}^2 + \frac{1}{vm} \sum_{j=1}^{m} \xi_j - \rho, \tag{6}$$

subject to (for j = 1,...,m)

$$\langle w(\cdot), k(x_j, \cdot) \rangle_{\chi} \geqslant \rho - \xi_j, \tag{7}$$

where $\xi_j \geqslant 0$, v is a control parameter for the fraction of the data that are allowed to be located on the wrong side of the $\mathcal{W}$ (outliers which do not belong to the $\mathcal{R}_{\chi}$) and ξ_j are slack variables.

It can be shown that a solution to Eq. (6,7) can be expressed by the following:

$$w(\cdot) = \sum_{j=1}^{m} \alpha_j k(x_j, \cdot), \tag{8}$$

where α_j comes from the dual optimization problem

$$\min_{\alpha} \frac{1}{2} \sum_{j,j'=1}^{m} \alpha_j \alpha_{j'} k(x_j, x_{j'}), \tag{9}$$

subject to $0 \leq \alpha_j \leq \frac{1}{vm}, \sum_j \alpha_j = 1$.

The OCSVM decision function $f_{\chi}(x)$ is given as follows:

$$f_{\chi}(x) = \sum_{j}^{m} \alpha_j k(x_j, x) - \rho, \tag{10}$$

where the value of ρ is calculated from knowing that $f_{\chi}(x_j) = 0$ for those $x_j \in \chi$ that verify both $\alpha_j \neq 0$ and $\alpha_j \neq \frac{1}{vm}$. Objects from χ that satisfies those conditions are located onto a decision boundary.

3 Proposed Approach

In this paper we study the one-class recognition task as a decomposition of single multi-class recognition problem. Instead of a binary classifier, which decomposes M classes into several binary problems, we use only patterns from target class to train the classifier.

The paper presents a novel approach for constructing OCC ensembles. The main aim of this paper is to discuss, if increasing the number of OCC per class will result in improvement of the classification performance. Canonical approaches for M-class problems assume that a pool of simple classifiers consists of M one-class classifiers, one per each target class. Therefore the behavior of a ensemble consisting of more than one OCC per class is worth investigating.

We propose an evolutionary scheme for selecting OCC from the pool of simple classifiers. This allows us to create a flexible meta-learning algorithm for OCC ensemble creation, which would not put any constrains on the number of classifiers per class. Additionally we use a simple, yet effective diversity measure to assure the heterogeneity in the multiple classifier system. We compare the presented approach with the canonical OCC ensemble(consisting of one OCC per class) to investigate if increasing the number of OCC per class is a worthwhile direction of research.

3.1 A Pool of Individual One-Class Classifiers

The pool is a set of K individual OCCs:

$$\Pi^{\Psi} = \{\Psi_1^1, \Psi_2^1, ..., \Psi_{i_1}^1, \Psi_1^2, ..., \Psi_{i_{M-1}}^{M-1}, \Psi_1^M, ..., \Psi_{i_M}^M\}, \tag{11}$$

where M stands for the number of classes in the problem under consideration, i_m stands for $i-$th classifier trained on a $m-$th class and $\sum_{m=1}^{M} \sum_{i=1}^{i_m} = K$.

In this paper we propose to verify if increasing the number of OCCs per class may lead to the improvement of classification accuracy. Therefore to create a pool of base classifier and at the same time assure its reasonable diversity we propose to use a random subspace method.

Random subspace method (or attribute bagging) [13] is a generalization of the random forest algorithm. Whereas random forests are composed of fully-grown decision trees, a random subspace classifier can be created with the usage of any underlying classifiers. Random subspace method has been used for various combinations of decision trees, linear classifiers, support vector machines and other types of classifiers. It consists of three main steps : adjusting the number of base classifiers, adjusting the number of subspaces and randomly selecting features for each of those subspaces, on which the classifiers are trained. This allows to increase the diversity of the ensemble.

For each of the $m \in \boldsymbol{M}$ classes the proposed approach creates i_m-1 subspaces and trains a simple classifier on them. The last classifier from the i_m is a classifier trained on all available patterns from $m-$th class. This allows the possibility of choosing a single classifier, instead of predictors trained on subspaces.

As a reference model we propose a canonical OCC ensemble, where for M classes we train the same number of classifiers i.e., each of them on all available patterns from corresponding class.

3.2 Classifier Selection

For selecting individual classifiers for the ensemble, we employ a genetic algorithm (GA). The chromosome Ch represents a model of compound classifier parameters and is a structure consisting of M components::

$$Ch = [C_1, C_2, ..., C_M], \tag{12}$$

where for each $m \in \boldsymbol{M}$ vector $C_m = \{C_{m1}, C_{m2}, \ldots, C_{mi_m}\}$ represents the OCCs from the pool for $m-$th class. Any procedure that is to be performed on the chromosome will take into account the fact that its M parts have quite a different character they consist of OCCs assigned to different classes. Therefore we restrict that information will not be exchanged between the part of the chromosomes processed by genetic operators.

An individual in the GA population represents a classifier ensemble, and consists of a binary vector, with 1s indicating the chosen individual classifiers. For example if we have 3 classes and 9 one-class classifiers then '[001],[011],[101]' would signify that for first class classifier 1, for second class classifiers 2,3 and for third class classifiers 1,3 are chosen for the ensemble. In case where one of the components consists only of zeros (i.e. no classifier is chosen) we change the bit value randomly. This is due to the specificity of the OCC problem - there always should be at least one OCC per each class.

The control parameters of the genetic algorithm are as follows:

- N_c - the upper limit of algorithm cycles,
- N_p - the population quantity,
- β - the mutation probability,
- γ - the crossover probability,
- Δ_m - the mutation range factor,
- V - the upper limit of algorithm iterations without quality improvement.

The ensemble classification error, calculated on the training set, serves as fitness function. Termination conditions can in principle be adjusted, we usually use the number of iterations without result improvement.

3.3 Diversity Assurance

We would like to introduce an additional measure for assuring the diversity in presented ensemble creation algorithm. Random subspace is considered as a good heuristic for introducing diversity, but due to the randomness element it often does not work perfectly. Therefore we had implemented an additional mechanism for forcing the diversity in the ensemble.

We use a *disagreement measure*. This measure is based on the intuition that two diverse classifiers perform differently on the same training data. Given two simple classifiers Ψ_j and Ψ_k , let $n(a, b)$ be the number of training samples on which the oracle output of h_j and h_k is a and b respectively. The diversity between the two classifiers is measured by:

$$dis_{j,k} = \frac{n(1,-1) + n(-1,1)}{n(1,1) + n(-1,1) + n(1,-1) + n(-1,-1)}. \tag{13}$$

The diversity increases with the value of the disagreement measure. As *disagreement measure* was introduced for canonical multiclass problems we propose to tune it for the OCCs ensemble. We assume that:

$$\begin{gathered} dis_{j,k} = \max dis_{value} \\ \text{if } \Psi_j \mapsto m_1; \Psi_k \mapsto m_2; m_1 \bigcap m_2 = \emptyset. \end{gathered} \tag{14}$$

This means that the value of the disagreement measure is maximal when two compared OCCs are trained on different classes.

We propose to use this measure to ensure the diversity of classifiers before the crossover process. It takes a form of two-step tournament selection:

- estimating the pairwise diversity between all the candidate classifiers and discarding those with low value of the disagreement measure - by randomly removing from the chromosome one of the two similar classifiers,
- proceeding with a normal tournament selection on altered chromosomes.

Therefore by utilizing both random subspace method for a pool of individual classifier creation and disagreement measure for controlling the chromosomes before the crossover we ensure that ensembles created by our proposed approach are heterogeneous.

3.4 Classifiers Fusion

The Error-Correcting Output Codes (ECOC) [5] framework is a simple yet effective framework created for dealing with the multi-class categorization problem the reconstruction from the decisions of binary classifiers. The basis of the ECOC framework consists of designing a codeword for each of the classes. These codewords encode the membership information of each class. Arranging the codewords as rows of a matrix, we obtain a encoding matrix Mc. Each of these binary problems (or dichotomizers) splits the set of classes in two partitions (coded by +1 or -1 in Mc according to their class set membership or 0 if the class is not considered by the current binary problem). Then, at the decoding step, applying the n trained binary classifiers, a code is obtained for each data point in the test set. This code is compared to the base codewords of each class denned in the matrix Mc, and the data point is assigned to the class with the closest codeword. ECOC can be easily used for OCCs ensemble, as we can map the target class as +1 and the unknown, outlier class by -1.

For this problem we use a exhaustive codes generation method, which was shown to perform very good for problems with class number no greater than seven [23].

4 Experimental Evaluation

The main objective of the experiments was to examine the behavior of the proposed method and to compare it to the canonical approach for training OCCs on several classes.

4.1 Experimental Set-Up

All experiments were carried out in R language [22] and computer implementations of used classification and optimization methods were taken from dedicated packages build-in mentioned above software or were implemented by the authors. The parameters used for the evolutionary algorithm were set as follows: $N_c = 200$, $N_p = 100$, $\beta = 0.7$, $\gamma = 0.3$, $\Delta_m = 0.2$, and $V = 50$.

For testing, we used a statistical test to compare the results and judge if their differences were statistically significant. For this purpose, we used a combined 5 x 2 cv F Test [1], where the subspace creation and classifier selection was repeated for each of the folds.

Datasets were taken from [10]. Their characteristics are presented in the Table 1.

4.2 Results

The results of experimental investigations are presented in the Table 2. Statistically significant results are bolded.

In the Table 3 we have presented the number of OCCs chosen for each of the classes in cases, where our proposed approach has outperformed the canonical (one OCC per class) method.

Table 1. Statistics of the datasets used in the experiments

No	dataset	Objects	Features	Classes
1	Iris	150	4	3
2	Wine	178	13	3
3	Parkinsons	197	23	2
4	Pima Indians Diabetes	768	8	2
5	Breast Cancer	286	9	2
6	Ozone Level	2536	73	2
7	Vehicle Silhouettes	946	18	4
8	Splice-junction Gene Sequences	3190	61	3
9	Dermatology	366	33	6
10	Musk (Version 2)	6598	168	2

Table 2. Results of the experiment

No	Canonical	Diversity-based	No	Canonical	Diversity-based
1	92.750	92.750	6	58.670	**65.005**
2	89.975	89.975	7	72.900	73.350
3	85.000	**87.230**	8	56.498	**62.976**
4	62.350	62.350	9	77.150	**83.568**
5	85.235	**89.045**	10	64.750	66.026

Table 3. Number of simple one-class classifiers per class in cases where evolutionary approach outperformed the canonical one

No	Class1	Class2	Class3	Class4	Class5	Class6
3	3	1	·	·	·	·
5	1	4	·	·	·	·
6	2	4	·	·	·	·
7	1	2	2	1	·	·
8	3	3	1	·	·	·
9	1	2	2	2	2	1
10	2	1	·	·	·	·

4.3 Results Discussion

Results of the experimental evaluation give a clear indication that introducing diversity into the OCC fusion is beneficial for the overall accuracy of the ensemble. In seven out of ten cases increasing the number of simple OCCs per single class returned a significant gain in accuracy. In most cases the higher number of simple classifiers were chosen only for some of the classes, not for all of them. Still the overall accuracy of the ensemble gained both from introducing diversity and from increasing the number of classifiers. Some of the datasets, like *dermatology* have an unbalanced class distribution. In these cases the proposed approach always improved the final decision.

It is worth mentioning that our proposed approach is flexible - it does not force the higher number of OCCs where it is not necessary. It happened in three out of ten cases, where the best results were returned by a canonical approach. In such cases the presented method adopted easily to the nature of the data and turned into the canonical fusion method.

5 Conclusions and Future Work

In this paper two main issues were addressed: does increasing the number of simple OCCs per class leads to the reduction of predictive error and does such OCC ensembles require the diversity measures. Answer to both of the questions is yes. Experiments clearly stated that presented method is no worse than the canonical one and in most cases outperforms it. Proposed approach utilised the

evolutionary algorithm for flexible selection of simple classifiers form the pool. The advantage of this method is that it puts no restriction on the number of OCCs in the ensemble, so when required, it can adapt itself to the canonical OCCs ensemble approach.

Introducing the diversity into the OCCs pool by combining random subspace and novel modification of disagreement measure allowed to create heterogeneous committees and explore the individual strengths of simple classifiers.

This work proved that investigating the new methods for diversity assurance in OCC ensembles is a promising research track. As one-class approach varies from the typical pattern recognition, standard methods form this field may not always be applicable. Therefore in our future works we would like to introduce new, more sophisticated diversity measures designed specifically for OCCs. Additionally presented results indicated that this approach may be useful for dealing with the unbalanced datasets. Finally we would like to apply our OCC ensembles for real-life problems from the areas of medicine, bioinformatics, chemoinformatics and environmental engineering.

Acknowledgment. This work is supported in part by the Polish National Science Centre under a grant for the period 2011-2014.

References

1. Alpaydin, E.: Combined 5 x 2 cv f test for comparing supervised classification learning algorithms. Neural Computation 11(8), 1885–1892 (1999)
2. Bishop, C.M.: Novelty detection and neural network validation. IEE Proceedings: Vision, Image and Signal Processing 141(4), 217–222 (1994)
3. Burduk, R.: Costs-Sensitive Classification in Multistage Classifier with Fuzzy Observations of Object Features. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS (LNAI), vol. 6679, pp. 245–252. Springer, Heidelberg (2011)
4. Burduk, R., Kurzyñski, M.: Two-stage binary classifier with fuzzy-valued loss function. Pattern Analysis and Applications 9(4), 353–358 (2006)
5. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. J. Artif. Int. Res. 2, 263–286 (1995)
6. Duan, K., Keerthi, S.S., Chu, W., Shevade, S.K., Poo, A.N.: Multi-Category Classification by Soft-Max Combination of Binary Classifiers. In: Windeatt, T., Roli, F. (eds.) MCS 2003. LNCS, vol. 2709, pp. 125–134. Springer, Heidelberg (2003)
7. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley-Interscience (2001)
8. Duin, R.: The combining classifier: to train or not to train? In: Proceedings of 16th International Conference on Pattern Recognition, vol. 2, pp. 765–770 (2002)
9. van Erp, M., Vuurpijl, L., Schomaker, L.: An overview and comparison of voting methods for pattern recognition. In: Proceedings of Eighth International Workshop on Frontiers in Handwriting Recognition, pp. 195–200 (2002)
10. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
11. Giacinto, G., Perdisci, R., Del Rio, M., Roli, F.: Intrusion detection in computer networks by a modular ensemble of one-class classifiers. Inf. Fusion 9, 69–82 (2008)

12. Hashem, S., Schmeiser, B., Yih, Y.: Optimal linear combinations of neural networks: an overview. In: 1994 IEEE International Conference on Neural Networks, IEEE World Congress on Computational Intelligence, June-2 July, vol. 3, pp. 1507–1512 (1994)
13. Ho, T.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
14. Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review 22(2), 85–126 (2004)
15. Hong, J.H., Min, J.K., Cho, U.K., Cho, S.B.: Fingerprint classification using one-vs-all support vector machines dynamically ordered with naïve bayes classifiers. Pattern Recogn. 41, 662–671 (2008)
16. Jain, A., Duin, R., Mao, J.: Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 4–37 (2000)
17. Kittler, J., Alkoot, F.: Sum versus vote fusion in multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(1), 110–115 (2003)
18. Koch, M.W., Moya, M.M., Hostetler, L.D., Fogler, R.J.: Cueing, feature discovery, and one-class learning for synthetic aperture radar automatic target recognition. Neural Networks 8(7-8), 1081–1102 (1995)
19. Krzanowski, W., Partrige, D.: Software Diversity: Practical Statistics for its Measurement and Exploatation. Department of Computer Science, University of Exeter. (1996)
20. Kuncheva, L.: Combining pattern classifiers: Methods and algorithms. Wiley-Interscience, New Jersey (2004)
21. Passerini, A., Pontil, M., Frasconi, P.: New results on error correcting output codes of kernel machines. IEEE Transactions on Neural Networks 15(1), 45–54 (2004)
22. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008) ISBN 3-900051-07-0, http://www.R-project.org
23. Rokach, L.: Pattern classification using ensemble methods. Series in machine perception and artificial intelligence. World Scientific (2010)
24. Schölkopf, B., Smola, A.: Learning with kernels: support vector machines, regularization, optimization, and beyond. In: Adaptive Computation and Machine Learning. MIT Press (2002)
25. Tax, D.M.J., Duin, R.P.W.: Combining One-Class Classifiers. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 299–308. Springer, Heidelberg (2001)
26. Tax, D., Duin, R.P.W.: Characterizing one-class datasets. In: Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa, pp. 21–26 (2005)
27. Wilk, T., Wozniak, M.: Soft computing methods applied to combination of one-class classifiers. Neurocomput. 75, 185–193 (2012)
28. Xu, L., Krzyzak, A., Suen, C.: Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Transactions on Systems, Man and Cybernetics 22(3), 418–435 (1992)

Author Index