

Nicolás García-Pedrajas
Francisco Herrera
Colin Fyfe
José Manuel Benítez
Moonis Ali (Eds.)

LNAI 6097

Trends in Applied Intelligent Systems

23rd International Conference
on Industrial Engineering and Other Applications
of Applied Intelligent Systems, IEA/AIE 2010
Cordoba, Spain, June 2010, Proceedings, Part II

2
Part II

 Springer

Lecture Notes in Artificial Intelligence

6097

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

Subseries of Lecture Notes in Computer Science

Nicolás García-Pedrajas
Francisco Herrera Colin Fyfe
José Manuel Benítez Moonis Ali (Eds.)

Trends in Applied Intelligent Systems

23rd International Conference
on Industrial Engineering and Other Applications
of Applied Intelligent Systems, IEA/AIE 2010
Cordoba, Spain, June 1-4, 2010
Proceedings, Part II

Series Editors

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

Volume Editors

Nicolás García-Pedrajas
University of Cordoba, Dept. of Computing and Numerical Analysis
Campus Universitario de Rabanales, Einstein Building, 14071 Cordoba, Spain
E-mail: npedrajas@uco.es

Francisco Herrera
José Manuel Benítez
University of Granada, Dept. of Computer Science and Artificial Intelligence
ETS de Ingenierías Informática y de Telecomunicación, 18071 Granada, Spain
E-mail: {herrera,j.m.benitez}@decsai.ugr.es

Colin Fyfe
University of the West of Scotland, School of Computing
Paisley, PA1 2BE, UK
E-mail: colin.fyfe@uws.ac.uk

Moonis Ali
Texas State University-San Marcos, Department of Computer Science
601 University Drive, San Marcos, TX 78666-4616, USA
E-mail: ma04@txstate.edu

Library of Congress Control Number: 2010926289

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-13024-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-13024-3 Springer Berlin Heidelberg New York

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

springer.com

© Springer-Verlag Berlin Heidelberg 2010
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

The need for intelligent systems technology in solving real-life problems has been consistently growing. In order to address this need, researchers in the field have been developing methodologies and tools to develop intelligent systems for solving complex problems. The International Society of Applied Intelligence (ISAI) through its annual IEA/AIE conferences provides a forum for international scientific and industrial community in the field of Applied Artificial Intelligence to interactively participate in developing intelligent systems, which are needed to solve twenty first century's ever growing problems in almost every field.

The 23rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligence Systems (IEA/AIE-2010) held in Córdoba, Spain, followed IEA/AIE tradition of providing an international scientific forum for researchers in the field of applied artificial intelligence. The presentations of the invited speakers and authors mainly focused on developing and studying new methods to cope with the problems posed by real-life applications of artificial intelligence. Papers presented in the twenty third conference in the series covered theories as well as applications of intelligent systems in solving complex real-life problems.

We received 297 papers for the main track, selecting 119 of them with the highest quality standards. Each paper was revised by at least three members of the Program Committee. The papers in the proceedings cover a wide number of topics including: applications to robotics, business and financial markets, bioinformatics and biomedicine, applications of agent-based systems, computer vision, control, simulation and modeling, data mining, decision support systems, evolutionary computation and its applications, fuzzy systems and their applications, heuristic optimization methods and swarm intelligence, intelligent agent-based systems, internet applications, knowledge management and knowledge based systems, machine learning, neural network applications, optimization and heuristic search, and other real-life applications.

The main track was complemented with 13 special sessions whose topics included soft computing in information access systems on the web, data preprocessing in data mining, engineering knowledge and semantic systems, applied intelligent systems for future classrooms, soft computing methods for environmental and industrial applications, soft computing in computer vision and image processing, distributed problem solving with artificial intelligence techniques, ensemble learning, interactive and cognitive environments, context information in intelligent systems, data analysis, optimization and visualization for bioinformatics and neuroscience, industrial applications of data mining and semantic and linguistic visual information.

Together, these papers highlight new trends and frontiers of applied artificial intelligence and show how new research could lead to new and innovative

applications. They also show that new trends are appearing to cope with the increasingly difficult new challenges that are faced by artificial intelligence. We hope you will find them interesting and useful for your own research.

The conference also invited five outstanding scholars to give plenary keynote speeches. They were Nitesh Chawla, from the University of Notre Dame, USA, Óscar Cordon from the European Center for Soft Computing, Spain, Ludmila Kuncheva, from the University of Bangor, UK, José Luis Verdegay, from the University of Granada, Spain, and Pierre Rouzè, from Ghent University, Belgium.

We would like to express our thanks to the members of the Program Committee and all the reviewers of the special sessions for their hard work. This work is central to the success of any conference.

The conference was organized by the Research Group on Computational Intelligence and Bioinformatics of the University of Córdoba jointly with the Soft Computing and Intelligent Information Systems Research Group of the University of Granada in cooperation with the International Society of Applied Intelligence (ISAI).

We would like to thank all members of the organization for their unselfish efforts to make the conference a success. We also would like to thank the University of Córdoba and its Polytechnic School for their support. We would like to thank Springer for their help in publishing the proceedings. We would like to thank our main sponsors, ISAI, as well as our other sponsors: Association for the Advancement of Artificial Intelligence (AAAI), Association for Computing Machinery (ACM/SIGART), Canadian Artificial Intelligence Association (CAIAC), European Neural Network Society (ENNS), International Neural Network Society (INNS), Japanese Society for Artificial Intelligence (JSAI), Taiwanese Association for Artificial Intelligence (TAAI), Taiwanese Association for Consumer Electronics (TACE), and Texas State University-San Marcos.

We would like to thank the invited speakers for their interesting and informative talks of a world-class standard. We cordially thank all authors for their valuable contributions as well as the other participants in this conference. The conference would not have been possible without their support.

Thanks are also due to the many experts who contributed to making the event a success.

March 2009

Nicolás García-Pedrajas
Francisco Herrera
Colin Fyfe
José Manuel Benítez
Moonis Ali

Conference Organization

General Chair

Moonis Ali
Texas State University, San Marcos, Texas,
USA

Program Chairs

Colin Fyfe
Nicolás García-Pedrajas
Francisco Herrera
University of the West of Scotland, UK
University of Córdoba, Spain
University of Granada, Spain

Local Organizing Chair

César García-Osorio
University of Burgos, Spain

Special Session Chairs

José Manuel Benítez
Evelio González-González
University of Granada, Spain
University of La Laguna, Spain

Publicity Chair

Rafael Alcalá
University of Granada, Spain

Organizing Committee

Cecilio Angulo-Bahón
Bonifacio Castaño-Martin
Antonio Fernández-Caballero
Rafael del Castillo-Gomariz
Gonzalo Cerruela-García
Salvador García
César García-Osorio
Aida de Haro-García
Domingo Ortiz-Boyer
Jesús Maudés-Raedo
Carlos Pardo-Aguilar
Javier Pérez-Rodríguez
Juan Antonio Romero
del Castillo
Miguel Ángel Salido
Technical University of Catalonia
University of Alcalá
University of Castilla-La Mancha
University of Córdoba
University of Córdoba
University of Jaén
University of Burgos, Spain
University of Córdoba
University of Córdoba
University of Burgos
University of Burgos
University of Burgos
University of Córdoba
University of Córdoba
Technical University of Valencia

Special Sessions

1. Soft Computing in Information Access Systems on the Web
Enrique Herrera-Viedma, Antonio G. López-Herrera, Eduardo Peis and Carlos Porcel
2. Data Preprocessing in Data Mining
Jose A. Gámez and José M. Puerta
3. Engineering Knowledge and Semantic Systems (IWEKSS)
Jason J. Jung and Dariusz Król
4. Applied Intelligent Systems for Future Classroom
Jia-Ling Koh
5. Soft-Computing Methods for Environmental and Industrial Applications
Juan M. Corchado, Emilio S. Corchado and Dante I. Tapia
6. Soft Computing in Computer Vision/Image Processing
Edurne Barrenechea, Humberto Bustince, Pedro Couto and Pedro Melo-Pinto
7. Distributed Problem Solving with Artificial Intelligence Techniques
Miguel Á. Salido and Adriana Giret
8. Ensemble Learning: Methods and Applications
Juan J. Rodríguez and César García-Osorio
9. Interactive and Cognitive Environments
Cecilio Angulo and Juan Antonio-Ortega
10. Context Information in Intelligent Systems
José Manuel Molina López and Miguel Ángel Patricio
11. New Frontiers in Data Analysis, Optimization and Visualization for Bioinformatics and Neuroscience
Fazel Famili, José M. Peña, Víctor Robles and Ángel Merchán
12. Industrial Applications of Data Mining: New Paradigms for New Challenges
Cèsar Ferri Ramírez, José Hernández Orallo and María José Ramírez Quintana
13. Semantic and Linguistic Visual Information: Applications
Jesús Chamorro-Martínez and Daniel Sánchez

Invited Speakers

| | |
|--------------------|-------------------------------------------|
| Nitesh Chawla | University of Notre Dame, USA |
| Óscar Cordón | European Center for Soft Computing, Spain |
| Ludmila Kuncheva | University of Bangor, UK |
| José Luis Verdegay | University of Granada, Spain |
| Pierre Rouzè | Ghent University, Belgium |

Program Committee

Acosta Sánchez, L., Spain
Aguilar, J., Spain
Ajith, A., Norway
Alba, E., Spain
Bae, Y., South Korea
Bahamonde, A., Spain
Becerra-Alonso, D., Spain
Barbakh, W., Palestine
Belli, F., Germany
Bello, R., Cuba
Benavides Cuéllar, C., Spain
Bernadó-Mansilla, E., Spain
Borzemski, L., Poland
Bosse, T., The Netherlands
Brézillon, P., France
Bugarín, A. J., Spain
Bull, L., UK
Bustince, H., Spain
Caballero, Y., Cuba
Carse, B., UK
Carvalho, J. P. B., Portugal
Casillas, J., Spain
Castillo, Ò., Mexico
Chan, C. W., Hong Kong
Chan, Ch.-Ch., USA
Chang, Ch.-I., USA
Charles, D., UK
Chen, Sh.-M., Taiwan
Chien, B.-Ch., Taiwan
Chou, J.-H., Taiwan
Chung, P. W. H., UK
Coelho, A. L. V., Brazil
Corchado, E., Spain
Corchado, J. M., Spain
Cordón, Ó., Spain
Cornelis, C., Belgium
Cotta, C., Spain
Da Costa, J. M., Portugal
Dapigny, R., France
De Baets, B., Belgium
De Carvalho, A., Brazil
De Melo, P. J., Portugal
Del Jesús, M. J., Spain
Dreyfus, G., France
Esposito, F., Italy
Fatima, S., UK
Fernández, F., Spain
Ferri, F., Spain
Ferri, C., Spain
Gámez, J. A., Spain
García, S., Spain
Giráldez, R., Spain
Girolami, M., UK
Gomide, F., Brazil
Guesgen, H. W., New Zealand
Gutiérrez, P. A., Spain
Hagras, H., UK
Hendtlass, T., Australia
Herrera-Viedma, E., Spain
Hirota, K., Japan
Hong, T.-P., Taiwan
Hoogendoorn, M., The Netherlands
Huang, Y.-P., Taiwan
Hüllermeier, E., Germany
Hung, Ch.-Ch., USA
Hwang, G.-J., Taiwan
Ishibuchi, H., Japan
Ito, T., Japan
Jacquet, F., France
Kinoshita, T., Japan
Klawonn, F., Germany
Kumar, A. N., USA
Kumova, B. Í., Turkey
Larrañaga, P., Spain
Lee, Sh.-J., Taiwan
Lin, T. Y., USA
Llanes, O., Cuba
Loia, V., Italy
López Ibáñez, B., Spain
Lozano, J. A., Spain
Lozano, M., Spain
Ludermir, T. B., Brazil
Madani, K., France
Mahanti, P., Canada
Mansour, N., Lebanon
Marcelloni, F., Italy

Marichal Plasencia, N., Spain
 Martínez, L., Spain
 Matthews, M. M., USA
 Mehrotra, K. G., USA
 Meléndez, J., Spain
 Mizoguchi, R., Japan
 Molina, J. M., Spain
 Monostori, L., Hungary
 Murphey, Y. L., USA
 Nedjah, N., Brazil
 Nguyen, N. T., Poland
 Ohsawa, Y., Japan
 Okuno, H. G., Japan
 Olivas, J. Á., Spain
 Pan, J.-Sh., Taiwan
 Pedrycz, W., Canada
 Pelta, D., Spain
 Peña, J. M., Spain
 Peregrín, A., Spain
 Pereira de Souto, M. C., Brazil
 Prade, H., France
 R-Moreno, M. D., Spain
 Raj Mohan, M., India
 Ramaswamy, S., USA
 Rayward-Smith, V. J., UK
 Rivera, A. J., Spain
 Rodríguez, J. J., Spain
 Rojas, I., Spain
 Romero Zaliz, R., Spain
 Sadok, D. F. H., Brazil
 Sainz-Palmero, G., Spain
 Sánchez, D., Spain
 Sánchez-Marré, M., Spain
 Schetinin, V., UK
 Selim, H., Turkey
 Shpitalni, M., Israel
 Soomro, S., Austria
 Stützle, T., Germany
 Sun, J., UK
 Suzuki, K., Japan
 Tamir, D., USA
 Tan, A.-H., Singapore
 Tereshko, V., UK
 Thulasiram, R. K., Canada
 Tseng, L.-Y., Taiwan
 Tseng, V. SH.-M., Taiwan
 Valente de Oliveira, J., Portugal
 Valtorta, M., USA
 Vancza, J., Hungary
 Viharos, Z. J., Hungary
 Wang, L., Singapore
 Yang, Ch., Canada
 Yang, D.-L., Taiwan
 Yang, Y., China
 Yin, H., UK
 Zhang, Q., UK

Special Session Reviewers

| | | |
|----------------------|-----------------|----------------|
| Adeodato, P. | Berlanga, A. | Cavallaro, A. |
| Alonso, C. | Bermejo, P. | Chang, C. |
| Alonso, J. | Bielza, C. | Chen, B. |
| Alonso, R.S. | Boström, H. | Chen, L. |
| Alonso, S. | Bustamante, A. | Chiang, C. |
| Álvarez, L. | Cabestany, J. | Cilla, R. |
| Anguita, D. | Cabrerizo, F. | Corral, G. |
| Aranda-Corral, G. A. | Cao, L. | Costa, J. A. |
| Argente, E. | Carrasco, R. | Damas, S. |
| Arroyo Castillo, Á. | Carrascosa, C. | De Bra, P. |
| Bajo, J. | Casacuberta, F. | De la Cal, E. |
| Barber, F. | Castanedo, F. | De la Ossa, L. |
| Baruque, B. | Catala, A. | De Paz, J. F. |

| | | |
|-------------------------|-----------------------|----------------------|
| Del Valle, C. | Kokol, P. | Romero, F. P. |
| Dorronsoró, J. | Ku, W. | Ruíz, R. |
| Duro, R. | Kuncheva, L. | Rumí, R. |
| Escalera, S. | Lachice, N. | Salmerón, A. |
| Escot, D. | Latorre, A. | Sánchez, J. |
| Esteva, M. | Lorkiewicz, W. | Santana, R. |
| Euzenat, J. | Luis, Á. | Schmid, U. |
| Fang, C. | Malutan, R. | Sedano, J. |
| Fauteux, F. | Martín-Bautista, J. | Seepold, R. |
| Fernández Caballero, A. | Martínez, A. | Serrano-Guerrero, J. |
| Fernández-Luna, J. | Martínez-Madrid, N. | Sevillano, X. |
| Fernández-Olivares, J. | Mateo, J. | Shan, M. |
| Fernández, J. | Medina, J. | Simic, D. |
| Flach, P. | Menasalvas, E. | Soares, C. |
| Flores, J. | Micó, L. | Soto Hidalgo, J. |
| Frías-Martínez, E. | Montes, J. | Stiglic, G. |
| García Varea, I. | Morales, J. | Stoermer, H. |
| García, J. | Morente, F. | Tapia, E. |
| Gasca, R. | Muelas, S. | Tchagang, A. |
| Godoy, D. | Nebot, A. | Tortorella, F. |
| Gómez-Verdejo, V. | Olivares, J. | Valentini, G. |
| Gómez-Vilda, P. | Peis, E. | Van den Poel, D. |
| González-Abril, L. | Petrakieva, L. | Varela, R. |
| González, S. | Phan, S. | Vela, C. |
| Graña, M. | Porcel, C. | Velasco, F. |
| Guadarrama, S. | Poyatos, D. | Vellido, A. |
| Guerra, L. | Prados Suárez, B. | Ventura, S. |
| Han, Y. | Pujol, O. | Victor, P. |
| Herrero, A. | Rauterberg, M. | Villar, J. |
| Herrero, P. | Regazzoni, C. | Wozniak, M. |
| Hou, W. | Ribeiro, B. | Zafra, A. |
| Iñesta, J. M. | Rinner, B. | Zhang, Ch. |
| Jatowt, A. | Rodríguez Aguilar, J. | Zhang, M.-L. |
| Julián, V. | Rodríguez, A. | |
| Jurado, A. | Rodríguez, L. | |
| Kang, S. | Rodríguez, S. | |

Additional Reviewers

| | | |
|--------------------|-----------------------|------------------|
| Al-Shukri, S. | Chamorro-Martínez, J. | Di Mauro, N. |
| Appice, A. | Chen, N. | Fanizzi, N. |
| Aziz, A. | Chen, W. | Fernández, A. |
| Barrenechea, E. | Coheur, L. | Galar, M. |
| Carmona-Poyato, Á. | Couto, P. | García, Ó. |
| Ceci, M. | D'Amato, C. | Gaspar-Cunha, A. |

| | | |
|------------------------|----------------------|----------------------|
| Hajiany, A. | Martínez-Álvarez, F. | Romero, F. |
| Hernández-Orallo, J. | McKenzie, A. | Sanz, J. |
| Hurtado Martín, G. | Medina-Carnicer, R. | Serrano-Guerrero, J. |
| Hwang, G. | Montero, J. | Shu, F. |
| Iglesias Rodríguez, R. | Mucientes, M. | Sudarsan, S. |
| Jaffry, S. | Nakadai, K. | Szmidt, E. |
| Jiang, X. | Nepomuceno, I. | Teng, T. |
| Jung, E. | Pagola, M. | Umair, M. |
| Jurío, A. | Palomar, R. | Van Lambalgen, R. |
| Kang, Y. | Paternain, D. | Van Wissen, A. |
| Klein, M. | Peña, J. | Varela, R. |
| Komatani, K. | Pontes, B. | Vasudevan, B. |
| Leng, J. | Pontier, M. | Vázquez, F. |
| Lo, D. | Pradera, A. | Villanueva, A. |
| López-Molina, C. | Re, M. | Wang, Y. |
| Márquez, F. | Rodríguez, R. | Yamamoto, K. |

Table of Contents – Part II

Engineering Knowledge and Semantic Systems

| | |
|----------------------------------------------------------------------------------------------------------------------------|----|
| Improving Effectiveness of Query Expansion Using Information Theoretic Approach | 1 |
| <i>Hazra Imran and Aditi Sharan</i> | |
| Defining Coupling Metrics among Classes in an OWL Ontology | 12 |
| <i>Juan García, Francisco García, and Roberto Therón</i> | |
| Enterprise 2.0 and Semantic Technologies for Open Innovation Support | 18 |
| <i>Francesco Carbone, Jesús Contreras, and Josefa Z. Hernández</i> | |
| Algorithmic Decision of Syllogisms | 28 |
| <i>Bora İ. Kumova and Hüseyin Çakır</i> | |
| Matching Multilingual Tags Based on Community of Lingual Practice from Multiple Folksonomy: A Preliminary Result | 39 |
| <i>Jason J. Jung</i> | |

Ensemble Learning: Methods and Applications

| | |
|-----------------------------------------------------------------------------------------------------------------------------------|----|
| Multiclass Mineral Recognition Using Similarity Features and Ensembles of Pair-Wise Classifiers | 47 |
| <i>Rimantas Kybartas, Nurdan Akhan Baykan, Nihat Yilmaz, and Sarunas Raudys</i> | |
| Ensembles of Probability Estimation Trees for Customer Churn Prediction | 57 |
| <i>Koen W. De Bock and Dirk Van den Poel</i> | |
| Evolving Ensembles of Feature Subsets towards Optimal Feature Selection for Unsupervised and Semi-supervised Clustering | 67 |
| <i>Mihaela Elena Breaban</i> | |
| Building a New Classifier in an Ensemble Using Streaming Unlabeled Data | 77 |
| <i>Mehmed Kantardzic, Joung Woo Ryu, and Chamila Walgampaya</i> | |
| Random Projections for SVM Ensembles | 87 |
| <i>Jesús Maudes, Juan José Rodríguez, César García-Osorio, and Carlos Pardo</i> | |

| | |
|---------------------------------------------------------------------------------------------------------|-----|
| Rotation Forest on Microarray Domain: PCA versus ICA | 96 |
| <i>Carlos J. Alonso-González, Q. Isaac Moro-Sancho, Iván Ramos-Muñoz, and M. Aránzazu Simón-Hurtado</i> | |
| An Empirical Study of Multilayer Perceptron Ensembles for Regression Tasks | 106 |
| <i>Carlos Pardo, Juan José Rodríguez, César García-Osorio, and Jesús Maudes</i> | |
| Ensemble Methods and Model Based Diagnosis Using Possible Conflicts and System Decomposition | 116 |
| <i>Carlos J. Alonso-González, Juan José Rodríguez, Óscar J. Prieto, and Belarmino Pulido</i> | |

Evolutionary Computation and Applications

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Entropy-Based Evaluation Relaxation Strategy for Bayesian Optimization Algorithm | 126 |
| <i>Hoang N. Luong, Hai T.T. Nguyen, and Chang Wook Ahn</i> | |
| A New Artificial Immune System for Solving the Maximum Satisfiability Problem | 136 |
| <i>Abdesslem Layeb, Abdel Hakim Deneche, and Souham Meshoul</i> | |
| Power-Aware Multi-objective Evolutionary Optimization for Application Mapping on NoC Platforms | 143 |
| <i>Marcus Vinícius Carvalho da Silva, Nadia Nedjah, and Luiza de Macedo Mourelle</i> | |
| A Discrete Differential Evolution Algorithm for Solving the Weighted Ring Arc Loading Problem | 153 |
| <i>Anabela Moreira Bernardino, Eugénia Moreira Bernardino, Juan Manuel Sánchez-Pérez, Juan Antonio Gómez-Pulido, and Miguel Angel Vega-Rodríguez</i> | |
| A Parallel Genetic Algorithm on a Multi-Processor System-on-Chip | 164 |
| <i>Rubem Euzébio Ferreira, Luiza de Macedo Mourelle, and Nadia Nedjah</i> | |
| The Influence of Using Design Patterns on the Process of Implementing Genetic Algorithms | 173 |
| <i>Urszula Markowska-Kaczmar and Filip Krygowski</i> | |

Fuzzy Systems and Applications

| | |
|------------------------------------------------------------------|-----|
| Obtaining Significant Relations in L-Fuzzy Contexts | 183 |
| <i>Cristina Alcalde, Ana Burusco, and Ramón Fuentes-González</i> | |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Knowledge Extraction Based on Fuzzy Unsupervised Decision Tree: Application to an Emergency Call Center | 193 |
| <i>Francisco Barrientos and Gregorio Sainz</i> | |
| Optimization of Embedded Fuzzy Rule-Based Systems in Wireless Sensor Network Nodes | 203 |
| <i>M.A. Gadeo-Martos, J.A. Fernández-Prieto, J. Canada Bago, and J.R. Velasco</i> | |
| An Algorithm for Online Self-organization of Fuzzy Controllers | 212 |
| <i>Ana Belén Cara, Héctor Pomares, and Ignacio Rojas</i> | |
| A Mechanism of Output Constraint Handling for Analytical Fuzzy Controllers | 222 |
| <i>Piotr M. Marusak</i> | |
| Analysis of the Performance of a Semantic Interpretability-Based Tuning and Rule Selection of Fuzzy Rule-Based Systems by Means of a Multi-Objective Evolutionary Algorithm | 228 |
| <i>María José Gacto, Rafael Alcalá, and Francisco Herrera</i> | |
| Testing for Heteroskedasticity of the Residuals in Fuzzy Rule-Based Models | 239 |
| <i>José Luis Aznarte M. and José M. Benítez</i> | |
| Heuristic Methods and Swarm Intelligence for Optimization | |
| Heuristic Methods Applied to the Optimization School Bus Transportation Routes: A Real Case | 247 |
| <i>Luzia Vidal de Souza and Paulo Henrique Siqueira</i> | |
| Particle Swarm Optimization in Exploratory Data Analysis | 257 |
| <i>Ying Wu and Colin Fyfe</i> | |
| Using the Bees Algorithm to Assign Terminals to Concentrators | 267 |
| <i>Eugénia Moreira Bernardino, Anabela Moreira Bernardino, Juan Manuel Sánchez-Pérez, Juan Antonio Gómez-Pulido, and Miguel Angel Vega-Rodríguez</i> | |
| Multicriteria Assignment Problem (Selection of Access Points) | 277 |
| <i>Mark Sh. Levin and Maxim V. Petukhov</i> | |
| Composite Laminates Buckling Optimization through Lévy Based Ant Colony Optimization | 288 |
| <i>Roberto Candela, Giulio Cottone, Giuseppe Fileccia Scimemi, and Eleonora Riva Sanseverino</i> | |

| | |
|--------------------------------------------------------------------------------------------------------------------------------|-----|
| Teaching Assignment Problem Solver | 298 |
| <i>Ali Hmer and Malek Mouhoub</i> | |
| Swarm Control Designs Applied to a Micro-Electro-Mechanical Gyroscope System (MEMS) | 308 |
| <i>Fábio Roberto Chavarette, José Manoel Balthazar, Ivan Rizzo Guilherme, and Orlando Saraiva do Nascimento Junior</i> | |

Industrial Applications of Data Mining: New Paradigms for New Challenges

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| A Representation to Apply Usual Data Mining Techniques to Chemical Reactions | 318 |
| <i>Frank Hoonakker, Nicolas Lachiche, Alexandre Varnek, and Alain Wagner</i> | |
| Incident Mining Using Structural Prototypes | 327 |
| <i>Ute Schmid, Martin Hofmann, Florian Bader, Tilmann Häberle, and Thomas Schneider</i> | |
| Viability of an Alarm Predictor for Coffee Rust Disease Using Interval Regression | 337 |
| <i>Oscar Luaces, Luiz Henrique A. Rodrigues, Carlos Alberto Alves Meira, José R. Quevedo, and Antonio Bahamonde</i> | |
| Prediction of Web Goodput Using Nonlinear Autoregressive Models | 347 |
| <i>Maciej Drwal and Leszek Borzemski</i> | |
| Domain Driven Data Mining for Unavailability Estimation of Electrical Power Grids | 357 |
| <i>Paulo J.L. Adeodato, Petrônio L. Braga, Adrian L. Arnaud, Germano C. Vasconcelos, Frederico Guedes, Hélio B. Menezes, and Giorgio O. Limeira</i> | |

Intelligent Agent-Based Systems

| | |
|-------------------------------------------------------------------------------------|-----|
| Social Order in Hippocratic Multi-Agent Systems | 367 |
| <i>Ludivine Crépin, Yves Demazeau, Olivier Boissier, and François Jacquenet</i> | |
| Building an Electronic Market System | 377 |
| <i>Elaine Lawrence and John Debenham</i> | |
| Information Theory Based Intelligent Agents | 387 |
| <i>Elaine Lawrence and John Debenham</i> | |

| | |
|---------------------------------------------------------------------------|-----|
| A Possibilistic Approach to Goal Generation in Cognitive Agents | 397 |
| <i>Célia Da Costa Pereira and Andrea G.B. Tettamanzi</i> | |
| Modelling Greed of Agents in Economical Context | 407 |
| <i>Tibor Bosse, Ghazanfar F. Siddiqui, and Jan Treur</i> | |
| Modeling and Verifying Agent-Based Communities of Web Services | 418 |
| <i>Wei Wan, Jamal Bentahar, and Abdessamad Ben Hamza</i> | |

Interactive and Cognitive Environments

| | |
|-----------------------------------------------------------------------------------------------------------------|-----|
| An Ambient Intelligent Agent Model Based on Behavioural Monitoring and Cognitive Analysis | 428 |
| <i>Alexei Sharpanskykh and Jan Treur</i> | |
| The Combination of a Causal and Emotional Learning Mechanism for an Improved Cognitive Tutoring Agent | 438 |
| <i>Usef Faghihi, Philippe Fouriner-viger, Roger Nkambou, and Pierre Poirier</i> | |
| Driver's Behavior Assessment by On-board/Off-board Video Context Analysis | 450 |
| <i>Lorenzo Ciardelli, Andrea Beoldo, Francesco Pasini, and Carlo Regazzoni</i> | |
| An eHealth System for a Complete Home Assistance | 460 |
| <i>Jaime Martín, Mario Ibañez, Natividad Martínez Madrid, and Ralf Seepold</i> | |
| Tracking System Based on Accelerometry for Users with Restricted Physical Activity | 470 |
| <i>L.M. Soria-Morillo, Juan Antonio Álvarez-García, Juan Antonio Ortega, and Luis González-Abril</i> | |

Internet Applications

| | |
|--------------------------------------------------------------------------------------------------|-----|
| Web Query Reformulation Using <i>Differential Evolution</i> | 484 |
| <i>Prabhat K. Mahanti, Mohammad Al-Fayoumi, Soumya Banerjee, and Feras Al-Obeidat</i> | |
| On How Ants Put Advertisements on the Web | 494 |
| <i>Tony White, Amirali Salehi-Abari, and Braden Box</i> | |
| Mining Association Rules from Semantic Web Data | 504 |
| <i>Victoria Nebot and Rafael Berlanga</i> | |
| Hierarchical Topic-Based Communities Construction for Authors in a Literature Database | 514 |
| <i>Chien-Liang Wu and Jia-Ling Koh</i> | |

| | |
|----------------------------------------------------------------------------------|-----|
| Generating an Event Arrangement for Understanding News Articles on the Web | 525 |
| <i>Norifumi Hirata, Shun Shiramatsu, Tadachika Ozono, and Toramatsu Shintani</i> | |

| | |
|-----------------------------------------------------------------------------------------------------------|-----|
| Architecture for Automated Search and Negotiation in Affiliation among Community Websites and Blogs | 535 |
| <i>Robin M.E. Swezey, Masato Nakamura, Shun Shiramatsu, Tadachika Ozono, and Toramatsu Shintani</i> | |

Knowledge Management and Knowledge Based Systems

| | |
|------------------------------------------------------------------------|-----|
| Effect of Semantic Differences in WordNet-Based Similarity Measures... | 545 |
| <i>Raúl Ernesto Menéndez-Mora and Ryutaro Ichise</i> | |

| | |
|------------------------------------------------------------------------------------------------|-----|
| An Ontological Representation of Documents and Queries for Information Retrieval Systems | 555 |
| <i>Mauro Dragoni, Célia Da Costa Pereira, and Andrea G.B. Tettamanzi</i> | |

| | |
|-----------------------------------------------------------------------|-----|
| Predicting the Development of Juvenile Delinquency by Simulation..... | 565 |
| <i>Tibor Bosse, Charlotte Gerritsen, and Michel C.A. Klein</i> | |

| | |
|------------------------------------------------------------------------------------------------------------------------------------|-----|
| Building and Analyzing Corpus to Investigate Appropriateness of Argumentative Discourse Structure for Facilitating Consensus | 575 |
| <i>Tatiana Zidrasco, Shun Shiramatsu, Jun Takasaki, Tadachika Ozono, and Toramatsu Shintani</i> | |

| | |
|----------------------------------------------------------------------------------------------------------------------------|-----|
| Improving Identification Accuracy by Extending Acceptable Utterances in Spoken Dialogue System Using Barge-in Timing | 585 |
| <i>Kyoko Matsuyama, Kazunori Komatani, Toru Takahashi, Tetsuya Ogata, and Hiroshi G. Okuno</i> | |

| | |
|------------------------------------------------------------------------------|-----|
| A New Approach to Construct Optimal Bow Tie Diagrams for Risk Analysis | 595 |
| <i>Ahmed Badreddine and Nahla Ben Amor</i> | |

Machine Learning

| | |
|-------------------------------------------------------------------------------------------------------------|-----|
| Feature Selection and Occupancy Classification Using Seismic Sensors | 605 |
| <i>Arun Subramanian, Kishan G. Mehrotra, Chilukuri K. Mohan, Pramod K. Varshney, and Thyagaraju Damarla</i> | |

| | |
|--------------------------------------------------------------------------|-----|
| Extending Metric Multidimensional Scaling with Bregman Divergences | 615 |
| <i>Jigang Sun, Malcolm Crowe, and Colin Fyfe</i> | |

| | |
|-------------------------------------------------------------------------------------------------------------------------------------|------------|
| Independent Component Analysis Using Bregman Divergences | 627 |
| <i>Xi Wang and Colin Fyfe</i> | |
| Novel Method for Feature-Set Ranking Applied to Physical Activity Recognition | 637 |
| <i>Oresti Baños, Héctor Pomares, and Ignacio Rojas</i> | |
| Time Space Tradeoffs in GA Based Feature Selection for Workload Characterization | 643 |
| <i>Dan E. Tamir, Clara Novoa, and Daniel Lowell</i> | |
| Learning Improved Feature Rankings through Incremental Input Pruning for Support Vector Based Drug Activity Prediction | 653 |
| <i>Wladimiro Díaz-Villanueva, Francesc J. Ferri, and Vicente Cerverón</i> | |
| Scaling Up Feature Selection by Means of Democratization | 662 |
| <i>Aida de Haro-García and Nicolás García-Pedrajas</i> | |
| Author Index | 673 |

Improving Effectiveness of Query Expansion Using Information Theoretic Approach

Hazra Imran¹ and Aditi Sharan²

¹ Department of Computer Science

Jamia Hamdard, New Delhi, India

himran@jamiyahamdard.ac.in

² School of Computers and System Sciences

Jawaharlal Nehru University, New Delhi, India

aditisharan@mail.jnu.ac.in

Abstract. Automatic Query expansion is a well-known method to improve the performance of information retrieval systems. In this paper we have suggested information theoretic measures to improve efficiency of co-occurrence based automatic query expansion. We have used pseudo relevance feedback based local approach. The expansion terms were selected from the top N documents using co-occurrence based approach. They were then ranked using two different information theoretic approaches. First one is standard *Kullback-Leibler* divergence (KLD). As a second measure we have suggested use of a variant KLD. Experiments were performed on TREC-1 dataset. The result suggests that there is a scope of improving co-occurrence based query expansion by using information theoretic measures. Extensive experiments were done to select two important parameters: number of top N documents to be used and number of terms to be used for expansion.

Keywords: Automatic Query Expansion, Candidate Terms, Term Co-occurrence, Kullback-Leibler divergence, Relevance Feedback.

1 Introduction

Current information retrieval systems are limited by many factors reflecting the difficulty to satisfy user requirements expressed by short queries. Reformulation of the user queries is a common technique in information retrieval to cover the gap between the original user query and his need of information. The most widely used technique for query reformulation is query expansion, where the original user query is expanded with new terms extracted from different sources. Queries submitted by users are usually very short. Efthimiadis [7] has done a complete review on the classical techniques of query expansion. The main problem of query expansion is that in some cases the expansion process worsens the query performance. Improving the robustness of query expansion has been the goal of many researchers in the last years and most proposed approaches use external collections [8,9,10] to extract candidate terms for the expansion. In our previous work, [12] we have focused on how a thesaurus can be used for query expansion.

Query Expansion can be: Manual, semiautomatic and automatic. In corpus- based automatic query expansion the terms to be added to the query can either be selected globally (from the entire document collection) or locally (from top N retrieved documents). Methods based on global analysis are computationally very expensive and its effectiveness is not better than that of methods based on local analysis [32,15,16]. Xu and Croft [17] have suggested the use of local context analysis (LCA) to achieve tradeoff between local and global query expansion. Our work relates to automatic query expansion done locally.

Most of the automatic query expansion methods use co-occurrence based approach to select the terms for query expansion. However, this is very broad and general approach and all the co-occurring terms don't have equal probability of improving query performance. Therefore, some other measures must be used in order to filter out non-useful terms and select suitable terms. Selecting suitable query terms is only one step toward improving query performance. In order to optimize query performance some parameters are to be set: number of terms to be added to query, number of top ranked documents used for selecting query terms. In absence of any theoretical justifications these parameters have to be set empirically.

In this paper we have suggested some measures to improve efficiency of co-occurrence based query expansion. We have suggested use of information theoretic approaches to rank the co-occurring terms. One of the approaches used is Kullback-Liebler Divergence (KLD) and other is the variant of KLD. Extensive experiments have been done to adjust the parameters (number of terms to be added to query, number of top ranked documents). The results have been compared and analyzed for all the three methods.

In the rest of this paper, we first make a review on related work in Section 2. Sections 3 and 4 describe the co-occurrence and information-theoretic approaches, respectively; Section 5 describes our methodology. The experimental results are presented in Section 6 and Section 7 summarizes the main conclusions of this work.

2 Related Work

Early work of Maron[21] demonstrated the potential of term co-occurrence data for the identification of query term variants. Lesk[18] expanded a query by the inclusion of terms that had a similarity with a query term greater than some threshold value of the cosine coefficient. Lesk noted that query expansion led to the greatest improvement in performance, when the original query gave reasonable retrieval results, whereas, expansion was less effective when the original query had performed badly. Sparck Jones [30] has conducted the extended series of experiments on the ZOO-document subset of the Cranfield test collection. The terms in this collection were clustered using a range of different techniques and the resulting classifications were then used for query expansion. Sparck Jones results suggested that the expansion could improve the effectiveness of a best match searching, if only, the less frequent terms in the collection were clustered with the frequent terms being unclustered and if only, very similar terms were clustered together. This improvement in performance was challenged by Minker et al.[22].

Some work on query expansion has been based on probabilistic models of the retrieval process. Researchers have tried to relax some of the strong assumptions of a term statistical independence that normally needs to be invoked, if probabilistic retrieval models are to be used [4,26]. In a series of papers, Van Rijsbergen had advocated the use of query expansion techniques based on a minimal spanning tree (MST), which contains the most important of the inter-term similarities calculated using the term co-occurrence data and which is used for expansion by adding in those terms that are directly linked to query terms in the MST [13,29,2,31]. Later work compared relevance feedback using both expanded and nonexpanding queries and using both MST and non-MST methods for query expansion on the Vaswani test collection [28,29]. Voorhees [6] expanded queries using a combination of synonyms, hypernyms and hyponyms manually selected from WordNet, and achieved limited improvement on short queries. Stairmand[19] used WordNet for query expansion, but they concluded that the improvement was restricted by the coverage of the WordNet and no empirical results were reported. More recent studies focused on combining the information from both co-occurrence-based and handcrafted thesauri [24,25]. Liu et al.[27] used WordNet for both sense disambiguation and query expansion and achieved reasonable performance improvement. However, the computational cost is high and the benefit of query expansion using only WordNet is unclear. Carmel [5] measures the overlap of retrieved documents between using the individual term and the full query. Previous work [1] attempt to sort query terms according to the effectiveness based on a greedy local optimum solution. Ruch et al.[23] studied the problem in the domain of biology literature and proposed an argumentative feedback approach, where expanded terms are selected from only sentences classified into one of four disjunct argumentative categories. Cao [11] uses a supervised learning method for selecting good expansion terms from a number of candidate terms.

3 Co-occurrence Approach

The methods based on the term co-occurrence which have been used since the 70's to identify the semantic relationships that exist among terms. Van Rijsbergen [2] has given the idea of using co-occurrence statistics to detect the semantic similarity between terms and exploiting it to expand the user's queries. In fact, the idea is based on the Association Hypothesis:

“If an index term is good at discriminating relevant from non-relevant documents then any closely associated index term is likely to be good at this.”

The main problem with the co-occurrence approach was mentioned by Peat and Willet [14] who claim that similar terms identified by co-occurrence tend to occur also very frequently in the collection and therefore, these terms are not good elements to be discriminate between relevant and non-relevant documents. This is true when the co-occurrence analysis is done generally on the whole collection but if we, apply it only on the top ranked documents discrimination does occur to a certain extent. We have used the pseudo relevance feedback method where we select top N documents using cosine similarity measures and terms are selected from this set.

In order to select co-occurring terms we have used two well-know coefficients: - jaccard and frequency, which are as follows.

$$jaccard_co(t_i, t_j) = \frac{d_{ij}}{d_i + d_j - d_{ij}} \quad (1)$$

Where

d_i and d_j are the number of documents in which terms t_i and t_j occur, respectively, and d_{ij} is the number of documents in which t_i and t_j co-occur.

$$freq_co(t_i, t_j) = \sum_{d \in D} (f_{d,t_i} \times f_{d,t_j}) \quad (2)$$

$t_i =$ all terms of top N docs terms

$t_j =$ query terms

$f_{d,t_i} =$ frequency of term t_i in doc

$f_{d,t_j} =$ frequency of term t_j in doc

$d =$ top N doc

We apply these coefficients to measure the similarity between terms represented by the vectors. However, there is a risk in applying these measures directly, since the candidate term could co-occur with the original query terms in the top documents by chance. The higher its degree is in whole corpus, the more likely it is that candidate term co-occurs with query terms by chance. The larger the number of co-occurrences, the less likely that term co-occur with query terms by chance. In order to reduce probability of adding the term by chance, we use the following equation to measure the degree of co-occurrence of a candidate term with query.

$$co_degree(c, t_j) = \log_{10}(co(c, t_j) + 1) * (idf(c) / \log_{10}(D)) \quad (3)$$

Where

$$idf(c) = \log_{10}(N / N_c) \quad (4)$$

$N =$ number of documents in the corpus

$D =$ number of top ranked documents used

$c =$ candidate term listed for query expansion

$n_c =$ number of documents in the corpus that contain c

$co(c, t_j) =$ number of co-occurrences between c and t_j in the top ranked documents i. e. jaccard_co(t_i, t_j) or freq_co(t_i, t_j)

To obtain a value measuring how good c is for whole query Q , we need to combine its degrees of co-occurrence with all individual original query terms $t_1, t_2 \dots t_n$. For this suitabilityfor Q is computed.

$$SuitabilityforQ = f(c, Q) = \prod_{t_i \in Q} (\delta + co_degree(c, t_i))^{idf(t_i)} \quad (5)$$

To expand a query Q , we rank the terms in the top ranked documents according to their suitability for Q and choose the top ranked terms for query expansion.

In general the co-occurrence based approach selects highly frequent co-occurring terms with respect to the query terms. However, good query expansion terms are those terms that are closely related to the original query and are expected to be more frequent in the top ranked set of documents retrieved with the original query than in other subsets of the collection. Information theoretic approaches have been found useful to incorporate above-mentioned idea. Next section deals with use of information theoretic approach for query expansion.

4 Information-Theoretic Approach

Information theoretic approaches used in query expansion are based on studying the difference between the term distribution in the whole collection and in the subsets of documents that are relevant to the query, in order to, discriminate between good expansion terms and poor expansion term. One of the most interesting approaches based on term distribution analysis has been proposed by Claudio et al. [3], who uses the concept the Kullback-Liebler Divergence to compute the divergence between the probability distributions of terms in the whole collection and in the top ranked documents obtained using the original user query. The most likely terms to expand the query are those with a high probability in the top ranked set and low probability in the whole collection. For the term t this divergence is:

$$KLD(t) = [p_R(t) - p_C(t)] \log \frac{\frac{f(t)}{NR}}{p_C(t)} \quad (6)$$

Here $P_R(t)$ is the probability of t estimated from the corpus R . $P_C(t)$ is the probability of $t \in V$ estimated using the whole collection. To estimate $P_C(t)$, we used the ratio between the frequency of t in C and the number of terms in C , analogously to $P_R(t)$;

$$P_R(t) = \begin{cases} \gamma \frac{f(t)}{NR} & \text{if } t \in V(R) \\ \delta p_C(t) & \text{otherwise} \end{cases} \quad (7)$$

Where

c is the set of all documents in the collection

R is the set of top retrieved documents relative to a query.

$V(R)$ is the vocabulary of all the terms in R .

NR is the number of terms in R .

$f(t)$ is the frequency of t in R

We have done our experiments with one more variation in which we have used a function other than $f(t)/NR$, taking also into account the likely degree of relevance of the documents retrieved in the initial run:

$$KLD_variation(t) = [p_R(t) - p_c(t)] \log \frac{\frac{\sum_d f(t) \times score_d}{\sum_t \sum_d f(t) \times score_d}}{p_c(t)} \quad (8)$$

In order to see the effect of information theoretic measures, we first selected the expansion terms using suitability value (equation 5) then equation (6 and 8) was used to rank the selected terms. For calculating the value of $P_R(t)$ (equation 7) we set $\gamma=1$, which restricts the candidate set to the terms contained in R. and then the top ranked terms for query expansion.

5 Description of Our Methodology

We have performed local query expansion based on pseudo relevance feedback. Following are the steps in our methodology.

1. *Indexing* - Our system first identified the individual terms occurring in the document collection.
2. *Word stemming*. To extract word-stem forms, we used porter-stemming algorithm [20].
3. *Stop wording*. We used a stop list to delete the common occurring words from the documents.
4. *Document weighting*. We assigned weights to the terms in each document by the classical *tf.idf* scheme.
5. *Weighting of unexpanded query*: To weigh terms in unexpanded query, we used the *tf* scheme.
6. *Document ranking with unexpanded query*: We computed a document ranking using common coefficients jaccard between the document vectors and the unexpanded query vector.
7. *Listing of candidate terms*: We use *jacc_coefficient* or *freq_coefficient* using equation (1) or (2) to list out the candidate terms which could be used for expansion.
8. *Expansion term ranking*: The candidates were ranked by using equation (5) or (6) and top terms were chosen for expansion.
9. *Construction of expanded query*: We simply added the top terms to the original query.
10. *Document ranking with expanded query*: The final document ranking was computed by using jaccard coefficient between the document vectors and the expanded query vector.

6 Experiments

For our experiments, we used volume 1 of the *TIPSTER* document collection, a standard test collection in the IR community. Volume 1 is a 1.2 Gbyte collection of full-text articles and abstracts. The documents came from the following sources.

- WSJ -- Wall Street Journal (1986, 1987, 1988, 1989,1990,1991 and 1992)
- AP -- AP Newswire (1988,1989 and 1990)
- ZIFF -- Information from Computer Select disks (Ziff-Davis Publishing)
- FR -- Federal Register (1988)
- DOE -- Short abstracts from Department of Energy

We have used WSJ corpus, and TREC topic set, with 50 topics, of which we only used the title (of 2.3 average word length). In our first approach, equation (5) was used for selecting the expansion terms in ranked order. In the second approach, we selected all the terms based on suitability (equation (5)) (jaccard_coefficient is used to select the similar terms). These terms were then ranked using KLD measure (equation (6)). In a similar way, for the third approach we used a variant of KLD in order to select the subset of terms from the terms selected by suitability value. We have compared the result of all these approaches with that of unexpanded query.

We have used different measures to evaluate each method. The measures considered are MAP (Mean Average Precision), Precision@5, Precision@10, and R-Precision. Precision and Recall are general measures to quantify overall efficiency of a retrieval system. However, when a large number of relevant documents are retrieved overall precision and recall values do not judge quality of the result. A retrieval method is considered to be efficient if it has high precision at low recalls. In order to quantify this precision can be calculated at different recall levels. We have calculated Precision@5, Precision@10 recall level.

Parameter Study

We have studied two parameters that are fundamental in query expansion: number of candidate terms to expand the query and number of documents from the top ranked set used to extract the candidate terms. The optimal value of these parameters can be different for each method, and thus we have studied them for each case. Following graphs shows the result for different parameter values for each of the methods.

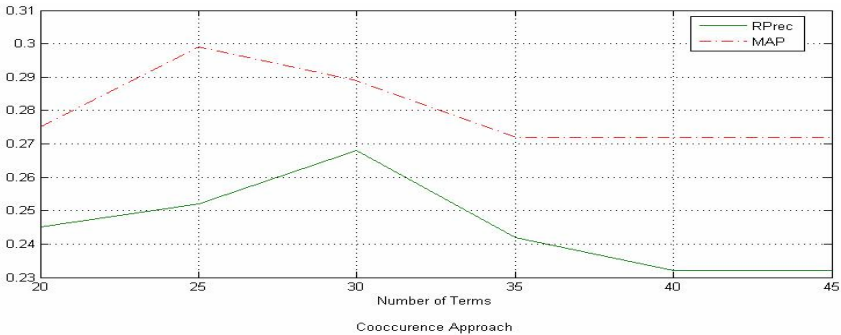


Fig. 1. Curve showing the MAP and R-PREC measures with different numbers of candidate terms to expand the original query using Co-occurrence Approach

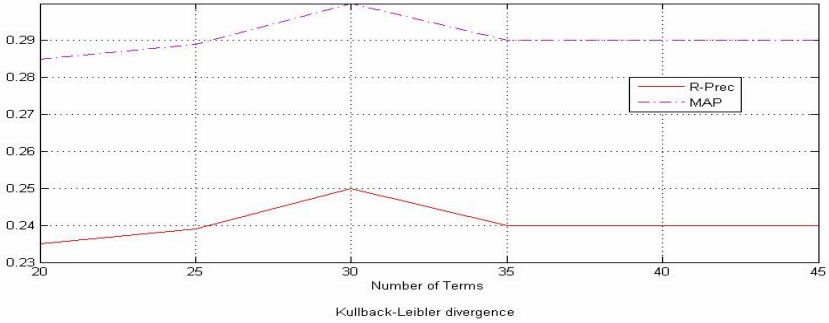


Fig. 2. Curve showing the MAP and R-PREC measures with different numbers of candidate terms to expand the original query using Kullback-Leibler divergence Approach

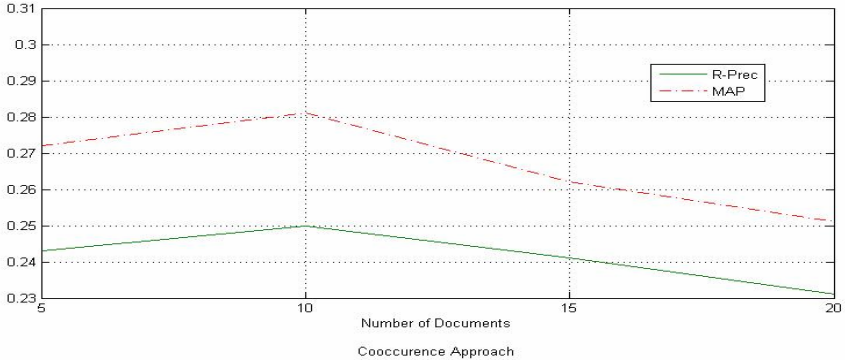


Fig. 3. Curve showing the MAP and R-PREC measures with different numbers of top documents used to extract the set of candidate query terms

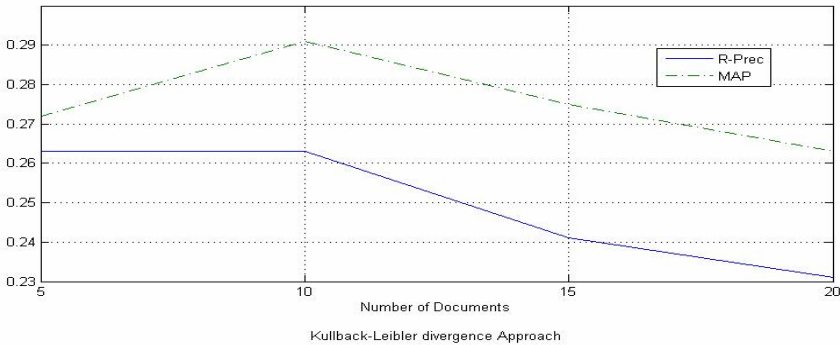


Fig. 4. Curve showing the MAP and R-PREC measures with different numbers of top documents used to extract the set of candidate query terms

We can observe that in all cases the best value for number of document selected for query expansion is around 10 documents and for the number of query expansion terms is 30. This implies that there is a certain threshold on number of documents and number of query expansion terms to be added in order to improve efficiency of query expansion.

Comparative Analysis of Result

Table 1 shows overall comparative result for all query expansion methods considered in our work. The parameter values for number of top documents is 10 and number of query terms to be added are 30. From the table we can observe that in general terms selected with suitability ranking are better candidates for query expansion in comparison to standard jaccard and frequency coefficients. We also observed that with the KLD we are able to improve the overall precision (MAP) and recall. In some cases, KLD_variant is able to improve precision@5. By changing various parameters, we may be able to visualize the effect of KLD_variant.

Table 1. Comparative result for query expansion methods used in our work. Best results appear in boldface.

| | MAP | P@5 | P@10 | R-Prec |
|-----------------------------------------------|--------------|--------------|--------------|--------------|
| Unexpanded query approach | .2413 | .3220 | .2915 | .2422 |
| Jaccard_coefficient | .2816 | .3450 | .2900 | .3102 |
| Freq_coefficient | .2218 | .3146 | .2995 | .3018 |
| Candidate term ranking using Suitability of Q | .2772 | .3660 | .2820 | .3643 |
| Candidate term ranking using KLD | .3012 | .3640 | .2860 | .3914 |
| KLD_variation | .2970 | .3665 | .2840 | .2802 |

7 Conclusions and Future Works

In this paper we have suggested the use of information theoretic measures in order to improve efficiency of co-occurrence based automatic query expansion. The experiments were performed on TREC dataset. We have used standard KLD as one of the information theoretic measures and suggested a variant of KLD. We observe that there is a considerable scope of improving co-occurrence based query expansion by using information theoretic measures. More experiments can be done in order to visualize the effect of suggested KLD variant. Further, the other information theoretic measures can be proposed to improve efficiency of automatic query expansion.

References

1. Lee, C.J., Lin, Y.C., Chen, R.C., Cheng, P.J.: Selecting effective terms for query formulation. In: Proc. of the Fifth Asia Information Retrieval Symposium (2009)
2. Van Rijsbergen, C.J.: A theoretical basis for the use of cooccurrence data in information retrieval. *Journal of Documentation* (33), 106–119 (1977)
3. Carpineto, C., Romano, G.: TREC-8 Automatic Ad-Hoc Experiments at Fondazione Ugo Bordoni, TREC (1999)
4. Croft, W.B., Harper, D.J.: Using probabilistic models of document retrieval without relevance information. *Journal of Documentation* 35, 285–295 (1979)
5. Carmel, D., Yom-Tov, E., Soboroff, I.: SIGIR Workshop Report: Predicting query difficulty – methods and applications. In: Proc. of the ACM SIGIR 2005 Workshop on Predicting Query Difficulty – Methods and Applications, pp. 25–28 (2005)
6. Voorhees, E.M.: Query expansion using lexical semantic relations. In: Proceedings of the 1994 ACM SIGIR Conference on Research and Development in Information Retrieval (1994)
7. Efthimiadis, E.N.: Query expansion. *Annual Review of Information Systems and Technology* 31, 121–187 (1996)
8. Voorhees, E.M.: Overview of the TREC 2003 robust retrieval track. In: TREC, pp. 69–77 (2003)
9. Voorhees, E.M.: The TREC 2005 robust track. *SIGIR Forum* 40(1), 41–48 (2006)
10. Voorhees, E.M.: The TREC robust retrieval track. *SIGIR Forum* 39(1), 11–20 (2005)
11. Cao, G., Nie, J.Y., Gao, J.F., Robertson, S.: Selecting good expansion terms for pseudo-relevance feedback. In: Proc. of 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 243–250 (2008)
12. Imran, H., Sharan, A.: Thesaurus and Query Expansion. *International journal of computer science & information Technology (IJCSIT)* 1(2), 89–97 (2009)
13. Harper, D.J., van Rijsbergen, C.J.: Evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation* 34, 189–216 (1978)
14. Peat, H.J., Willett, P.: The limitations of term co-occurrence data for query expansion in document retrieval systems. *JASIS* 42(5), 378–383 (1991)
15. Schütze, H., Pedersen, J.O.: A cooccurrence-based thesaurus and two applications to information retrieval. *Inf. Process. Manage* 33(3), 307–318 (1997)
16. Jing, Y., Croft, W.B.: An association thesaurus for information retrieval. In: 4th International Conference on Proceedings of RIAO 1994, New York, US, pp. 146–160 (1994)
17. Xu, J., Croft, W.B.: Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.* 18(1), 79–112 (2000)
18. Lesk, M.E.: Word-word associations in document retrieval systems. *American Documentation* 20, 27–38 (1969)
19. Stairmand, M.A.: Textual context analysis for information retrieval. In: Proceedings of the 1997 ACM SIGIR Conference on Research and Development in Information Retrieval (1997)
20. Porter, M.F.: An algorithm for suffix stripping. *Program - automated library and information systems* 14(3), 130–137 (1980)
21. Maron, M.E., Kuhns, J.K.: On relevance, probabilistic indexing and information retrieval. *Journal of the ACM* 7, 216–244 (1960)
22. Minker, J., Wilson, G.A., Zimmerman, B.H.: Query expansion by the addition of clustered terms for a document retrieval system. *Information Storage and Retrieval* 8, 329–348 (1972)

23. Ruch, P., Tbahriti, I., Gobeill, J., Aronson, A.R.: Argumentative feedback: A linguistically-motivated term expansion for information retrieval. In: Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pp. 675–682 (2006)
24. Mandala, R., Tokunaga, T., Tanaka, H.: Combining multiple evidence from different types of thesaurus for query expansion. In: Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval (1999)
25. Mandala, R., Tokunaga, T., Tanaka, H.: Ad hoc retrieval experiments using wordnet and automatically constructed thesauri. In: Proceedings of the seventh Text REtrieval Conference, TREC7 (1999)
26. Robertson, S.E., Sparck Jones, K.: Relevance weighting of search terms. *Journal of the American Society of Information Science* 21, 129–146 (1976)
27. Liu, S., Liu, F., Yu, C., Meng, W.: An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In: Proceedings of the 2004 ACM SIGIR Conference on Research and Development in Information Retrieval (2004)
28. Smeaton, A.F.: The retrieval effects of query expansion on a feedback document retrieval system, University College Dublin, MSc thesis (1982)
29. Smeaton, A.F., van Rijsbergen, C.J.: The retrieval effects of query expansion on a feedback document retrieval system. *Computer Journal* 26, 239–246 (1983)
30. Sparck Jones, K.: Automatic keyword classification for information retrieval. Butterworth, London (1971)
31. Van Rijsbergen, C.J., Harper, D.J., Porter, M.F.: The selection of good search terms. *Information Processing and Management* 17, 77–91 (1981)
32. Qiu, Y., Frei, H.-P.: Concept based query expansion. In: SIGIR, pp. 160–169 (1993)

Defining Coupling Metrics among Classes in an OWL Ontology

Juan García, Francisco García, and Roberto Therón

Department of Computer Science, University of Salamanca, Spain
{ganajuan,fgarcia,theron}@usal.es

Abstract. This paper aims to propose some new metrics to measure relationships among classes in an ontology. Relationships among classes in an OWL ontology are given by the object properties that are defined as a binary relation between classes in the domain with classes in the range. Our proposal is based on the coupling metric defined in the software engineering field adapted to an ontology needs. We have implemented and tested our metrics with real ontologies and the results are analysed and discussed.

Keywords: Ontology Metrics, Coupling between Classes, OWL Ontologies.

1 Introduction

In general, an ontology describes formally a domain of discourse. Typically, an ontology consists of a finite list of terms and the relationships between these terms. The terms denote important concepts (classes of objects) of the domain. Relationships typically include hierarchies of classes. A hierarchy specifies a class C to be a subclass of another class C' if every object in C is also included in C' . Apart from subclass relationships, ontologies may include information such as properties, value restrictions, disjoint statements, specifications of logical relationships between objects. We focus on OWL ontologies that distinguish between two main categories of properties: Object properties and Datatype properties. Object properties relate classes in the domain with classes in the range while Datatype properties relate classes in the domain with simple data values in the range. Relationships among classes are given by the Object properties so in order to measure them these properties need to be analysed. In software engineering, Coupling Between Objects (CBO) has been defined in [2] with two variants. CBO-in that measures the number of classes that depend on the class under consideration and CBO-out that measures the number of classes on which the class under consideration depends. This approach has been taken into account to define our metrics but focused on ontologies.

1.1 Our Contribution

In this paper we formally define the metrics that help to analyse and evaluate coupling between classes. These metrics are focused on providing an insight

about the importance of the classes according to their relationships and the way they are related. Moreover they are also useful to analyse large ontologies in order to simplify the identification of most coupled classes that represent the main classes.

We start this article with a brief introduction of ontologies and coupling, then we discuss some related work in the second part. In the third section we formally define our metrics proposal. The fourth section is used to analyse a case study with a real ontology and finally in the last section we conclude and discuss the future work.

2 Related Work

There is no so much work about metrics on ontologies. The paper [3] reviews the current state-of-the-art and basically proposes normalization as a pre-process to apply structural metrics. This normalization process consists of five steps: name anonymous classes, name anonymous individuals, classify hierarchically and unify the names, propagate the individuals to the deepest possible classes and finally normalize the object properties. This proposal is focused on content metrics based on OntoMetric framework and basically they have been proposed to improve ontology behaviour or to fix some mistakes. The paper [4] proposes some mechanisms to rank metrics to diverse ontologies. Basically this proposal consists of a Java Servlet to process as inputs some keywords introduced by the user. Then the framework searches using Swoogle¹ engine and retrieves all the URI's representing the ontologies related with these keywords. Then the framework searches on its internal database if these ontologies have been previously analysed and retrieves their information.

Without any doubt OntoQA [13][14] represents the main proposal about metrics on ontologies. It proposes some Schema Metrics to measure the richness of schema relationships, attributes and schema inheritance. These metrics are focused on evaluating the ontology in general. Another proposed categories are class richness, average population, cohesion, the importance of a class, fullness of a class, class inheritance and class relationship richness, connectivity and readability. This work describes two similar but not equal metrics. Class Relationship Richness is defined as the number of relationships that are being used by instances that belong to the class. By other hand, the Connectivity of a class is defined as the number of instances of other classes that are connected to instances of the selected class. The main differences are that these metrics take into account the instances belonging to the class instead of relations declared in the class.

Some papers related with coupling metrics on software engineering have been published [8][9][10]. One of the first papers to provide with definitions for coupling in object-oriented systems is [5]. Moreover it also analyses the effects of coupling and defines a coupling metric on an ordinal scale within the framework proposed. Authors take the previous definition of Wand and Weber [7] of CBO

¹ <http://swoogle.umbc.edu/>

(Coupling Between Objects); that was previously defined as a proportional value to the number of non-inheritance related couples with other classes. Based on this definition [6] proposes a new software complexity metrics based on five steps. The first one defines the calculation of modules coupling by counting the number of instances from one class inside other one and vice versa. The second one refers to the calculation of logical coupling while the third step refers to making clusters or sets of classes. The fourth step defines the intersection of two clusters and the final step calculates the average of the intersection rates. Furthermore a visualisation tool has been proposed to visualise the metrics.

3 Our Proposal

As we said above relations between classes in an ontology provide us with an insight of the ontology. The direction of a property can be logically considered going from the domain to the range. This direction provides us important information about the role that classes play in the ontology. For instance a class that belongs to the domain of many properties would represent one of the main subjects of the ontology. This is obvious because it implies that this class is being qualified, evaluated or described. In contrast, a class that belongs to the range of one or many properties would represent a qualifier, characteristic or even more important may be used to infer a certain type, and sometimes these classes represent an enumeration of values. It is important to differentiate between both types of classes in an ontology. Furthermore it is also important to distinguish between a property that has the same class belonging to both domain and range.

Our metrics have been defined to be the analogous counterpart from Object-Oriented Systems to ontologies. We define CBE (Coupling Between Entities) for ontologies with two possibilities. CBE-out representing the coupling where the class belongs to the domain of the property and CBE-in representing the coupling where the class belongs to the range of the property. Furthermore we also define SC (Self Coupling) for those properties that have the same class belonging to both domain and range of the property. Figure 1 shows a diagram where CBE is illustrated over class A. The same class may play different roles in an ontology, nevertheless almost always there are some of them that play specific roles being either subjects or qualifiers, depending on the side they belong to. As consequence we have formally defined 5 metrics that are listed below:

Definitions:

1. Let Θ be an OWL ontology
2. Let Φ be the set of properties $\in \Theta$
3. Let \mathbf{C} be the set of classes $\in \Theta$
4. $\exists c \in \mathbf{C}$
5. $\exists \rho \subseteq \Phi$

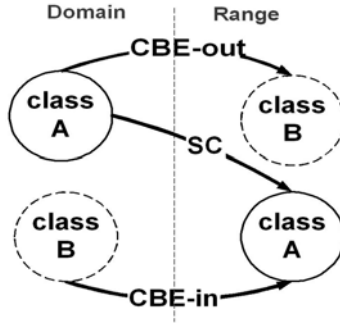


Fig. 1. Shows the CBE-out, CBE-in and SC definitions on class A

Metrics:

1. We define CBE-out metric as:
if $c \in \text{domain}(p) \forall p \in \rho$ then $\text{CBE-out} = |\rho|$
2. We define CBE-in metric as:
if $c \in \text{range}(p) \forall p \in \rho$ then $\text{CBE-in} = |\rho|$
3. We define CBE-io metric as:
if $\exists p \in \rho$ and $q \in \rho \mid p$ is inverse of q then $\text{CBE-io} = |\rho| / 2$
4. We define SC (Self Coupling) as:
if $\exists c \in \text{domain}(p)$ and $c \in \text{range}(p) \forall p \in \rho$ then $\text{SC} = |\rho|$
5. Finally we define the Total Coupling value as:

$$\text{TC} = \text{CBE-out} + \text{CBE-in} + \text{CBE-io} + \text{SC}$$

The main advantages of our metrics include: the capacity to detect classes that represent the subject of the ontology and classes that qualify, to be able to discern between both. Moreover metrics tell us information about how coupled classes in the ontology are, resulting in a low or high coupled ontology. We consider that high coupled ontologies would be desirable because low coupling would imply that classes are not related each other except for ISA relationships.

4 Case Study

According to our definition we implemented these metrics using Java Programming Language and Jena API² to manage OWL ontologies. We selected Semantic Web Technology Evaluation Ontology (SWETO) version 1.4³ to be analysed [12]. This general-purpose ontology was developed in 2004 by the Computer Science Department at the University of Georgia. This SWETO version consists of 114 classes and 13 object properties. Table 1 shows the result of applying our metrics to this ontology and all the classes that have at least 1 coupling value.

² <http://jena.sourceforge.net/>

³ <http://knoesis.wright.edu/library/ontologies/sweto/>

Table 1. Table 1 shows the coupling metrics for SWETO ontology

| Class | Object Properties | CBE-out | CBE-in | CBE-io | SC | CBE |
|------------------------------|-------------------|---------|--------|--------|----|-----|
| <i>Person</i> | 5 | 5 | 0 | 0 | 0 | 5 |
| <i>Organization</i> | 1 | 1 | 3 | 0 | 0 | 4 |
| <i>Publication</i> | 1 | 1 | 2 | 0 | 1 | 4 |
| <i>Place</i> | 0 | 0 | 3 | 0 | 0 | 3 |
| <i>AcademicDepartment</i> | 0 | 0 | 2 | 0 | 0 | 2 |
| <i>Event</i> | 1 | 1 | 1 | 0 | 0 | 2 |
| <i>Researcher</i> | 1 | 1 | 0 | 0 | 0 | 1 |
| <i>University</i> | 1 | 1 | 0 | 0 | 0 | 1 |
| <i>Professor</i> | 1 | 1 | 0 | 0 | 0 | 1 |
| <i>ScientificPublication</i> | 1 | 1 | 0 | 0 | 0 | 1 |
| <i>Thing</i> | 1 | 1 | 0 | 0 | 0 | 1 |
| <i>Country</i> | 0 | 0 | 1 | 0 | 0 | 1 |
| <i>Classification</i> | 0 | 0 | 1 | 0 | 0 | 1 |

The simple analysis of SWETO coupling metrics has shown interesting results. Even having no background information about the ontology, we can clearly deduce the most coupled class also represents the most important one, in this case 'Person'. This is an obvious result because if this class contains most of the object properties then it represents the main ontology's subject. Furthermore this fact can be supported by the fact that all the relationships are in the CBE-out metric which means that this class belongs to the domain in all the object properties. This fact implies these object properties were defined having as main purpose to satisfy some needs for this specific class. On the other hand, classes that only have CBE-in values such as Place, Academic Department, Country or Classification belong to the range in the properties. It means that individuals of these classes represent all the possible values the properties can take in the range. Moreover these individuals represent qualifiers for other entities. Analysing CBE-io metric we realise that there are no values greater than 0. It means there are no inverse functions declared in the ontology. Moreover there is only one class with self coupling value (Publication); meaning that exists one property which has the same class in the domain and range. This analysis let us to have an insight into the behaviour of the ontology even without previous knowledge background of it. The main purpose is to relate persons with places, organizations or academic departments. Furthermore more specialized classes such as Professor or Researcher are related with Scientific Publication. We can deduce from this ontology that it shows low coupling between classes. Just 13 out of 114 classes are related, which represents less than 10 percent of the total.

5 Conclusions and Future Work

We have provided a formal definition of some helpful metrics to analyse the coupling between classes in an ontology. These metrics are based in the coupling

metrics for software engineering field but modified to satisfy an ontology needs. We have analysed a case study using a public ontology focusing on the advantages of using our metrics. Our metrics let us discover the most significant classes in the ontology according to the way they interact with others as well as to have an insight of the role played by the individuals. Finally the future work will include the definition of more metrics as well as a visualisation to represent object and datatype properties, classes and coupling between them.

References

1. Antoniou, G., van Harmelen, F.: *A Semantic Web Primer*, 2nd edn. The MIT Press, Cambridge (2008)
2. Chidamber, S., Kemerer, C.: A Metrics suite for object Oriented Design. *IEEE Transactions on Software Engineering* 20(6), 476–493 (1994)
3. Vrandečić, D., Sure, Y.: How to Design Better Ontology Metrics. In: Franconi, E., Kifer, M., May, W. (eds.) *ESWC 2007*. LNCS, vol. 4519, pp. 311–325. Springer, Heidelberg (2007)
4. Alani, H., Brewster, C., Shadbolt, N.: Ranking Ontologies with AKTiveRank. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) *ISWC 2006*. LNCS, vol. 4273, pp. 1–15. Springer, Heidelberg (2006)
5. Hitz, M., Montazeri, B.: Measuring Coupling and Cohesion In Object-Oriented Systems. In: *Proceedings of the International Symposium on Applied Corporate Computing* (1995)
6. Hanakawa, N.: Visualization for software evolution based on logical coupling and module coupling. In: *14th Asia-Pacific Software Engineering Conference* (2007)
7. Wand, Y., Weber, R.: An Ontological Model of an Information System. *IEEE Trans. Software Engineering* (1990)
8. Allen, E., Khoshgoftaar, T.: Measuring Coupling and Cohesion: An Information-Theory Approach. In: *Sixth International Software Metrics Symposium, METRICS 1999* (1999)
9. Offutt, J., Harrold, M., Kolte, P.: A Software Metric System for Module Coupling. *Journal of Systems and Software* (1993)
10. Yang, H.Y., Tempero, E., Berrigan, R.: Detecting Indirect Coupling. In: *IEEE, Proceedings of the 2005 Australian conference on Software Engineering, ASWEC 2005* (2005)
11. Archer, C.: *Measuring Object-Oriented Software Products*. Carnegie Mellon University (1995)
12. Aleman-meza, B., Halaschek, C., Sheth, A., Arpinar, B., Sannapareddy, G.: SWETO: Large-Scale Semantic Web Test-bed. In: *16th International Conference on Software Engineering and Knowledge Engineering* (2004)
13. Tartir, S., Arpinar, B., Moore, M., Sheth, A., Aleman-meza, B.: *OntoQA: Metric-based ontology quality analysis*. CiteSeerX - Scientific Literature Digital Library and Search Engine (2005)
14. Tartir, S., Arpinar, B.: *Ontology Evaluation and Ranking using OntoQA*. In: *Proceedings of the International Conference on Semantic Computing* (2007)

Enterprise 2.0 and Semantic Technologies for Open Innovation Support

Francesco Carbone¹, Jesús Contreras¹, and Josefa Z. Hernández²

¹ iSOCO c/ Pedro de Valdivia 10, 28006, Madrid, Spain (s)

² Dept. of Artificial Intelligence, Technical Univ. of Madrid, Campus de Montegancedo s/n
28660 Boadilla del Monte, Madrid, Spain

{fcarbone, jcontreras}@isoco.com, phernan@fi.upm.es

Abstract. In recent years, Web 2.0 has achieved a key role in the Internet community and concepts such as “the wisdom of crowds” have grown in importance in the enterprise context. Companies are adopting this paradigm for their internal processes in the so-called Enterprise 2.0. Semantic technology seems to be essential for its successful adoption with a positive return of investment. On the other hand, the introduction of the Open Innovation model, for which the innovation process should be opened out of the R&D department to all the employees and external actors, requires a technological infrastructure to be supported. In this paper we discuss how the Web 2.0 philosophy and Semantic Technology support Open Innovation and how three big European companies have profited from this new paradigm.

Keywords: Enterprise 2.0, Semantic Web, Open Innovation, Semantic Technologies.

1 Introduction

In recent years computer science has faced more and more complex problems related to information creation and fruition. Applications in which small groups of users publish static information or perform complex tasks in a closed system are not scalable and nowadays are out of date. In the last years a new paradigm changed the way Web applications are designed and used: Web 2.0[1] represents ‘THE’ Web collaborative paradigm (according to O’Reilly and Battelle [2]) and can be seen as architecture of participation where users can contribute to the website content creation with network effects. The collaborative paradigm leads to the generation of large amounts of content and when a critical mass of documents is reached, information becomes unavailable. Knowledge and information management are not scalable unless formalisms are adopted. Semantic Web’s aim is to transform human readable content into machine readable [3]. With this goal data interchange formats (e.g. RDF/XML, N3, Turtle, N-Triples), and languages such as RDF Schema (RDFS) and the Web Ontology Language (OWL) have been defined.

Computer supported collaborative work research analyzed the introduction of Web 2.0 in corporations: McAfee [4] called “Enterprise 2.0” a paradigm shift in corporations towards the 2.0 philosophy: collaborative work should not be based in the hierarchical

structure of the organization but should follow the Web 2.0 principles of open collaboration. This is especially true for innovation processes when the open innovation paradigm is adopted (Chesbrough [5]). In a world of widely distributed knowledge, companies do not have to rely entirely on their own research, but should open the innovation to all the employees of the organization, to providers and customers.

In this paper we discuss how open innovation can be supported by Web 2.0 and semantic technologies demonstrating how these technologies increase efficiency and effectiveness. Three real systems have been implemented with this approach for three international corporations, in financial¹, energy² and telecommunication³ fields with an average of three thousands employees involved in the innovation processes of each firm.

2 Innovation Meets Enterprise 2.0 and Semantic Technologies

In a scenario in which collaborative work is not supported and members of the community can barely interact with others, solutions to everyday problems and organizational issues rely on individual initiative. Innovation and R&D management are complex processes for which collaboration and communication are fundamental. They imply creation, recognition and articulation of opportunities, which need to be evolved into a business proposition in a second stage. The duration of these tasks can be drastically shortened if ideas come not just from the R&D department. This is the basis of the open innovation paradigm which opens up the classical funnel to encompass flows of technology and ideas within and outside the organization.

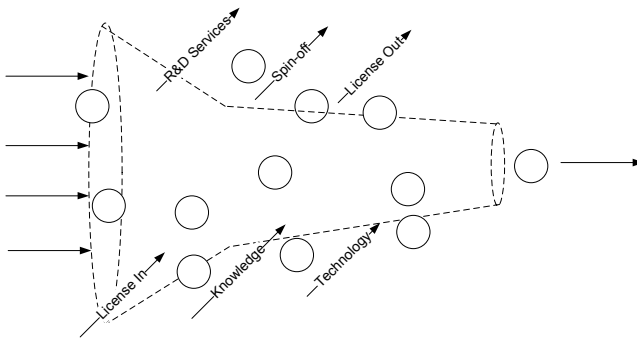


Fig. 1. Open Innovation

Ideas are pushed in and out the funnel until just a few reach the stage of commercialization. Technologies are needed to support the opening of the innovation funnel, to foster interaction for the creation of ideas (or patents) and to push them through and inside/outside the funnel. Platt [6] states that innovation, together with marketing and

¹ The customer is not revealed in any publication, but the entity is one of the 10 leading financial institution in Spain.

² Repsol www.repsol.com

³ Telefonica I+D www.tid.es

training, is one of the processes which could most benefit from the introduction of Enterprise 2.0 tools in a company. Gloor [7] defines the "Collaborative Innovation Networks" as "a cyberteam of self-motivated people with a collective vision, enabled by the Web to collaborate in achieving a common goal by sharing ideas, information, and work". The technology framework identified by Gloor has to grant a high degree of interactivity, connectivity and sharing. All these characteristics can be identified in an Enterprise 2.0 environment where editing and creating documents is easier and interaction and collaboration are key.

On the other hand, as we stated, performing analysis and structuring the information easily becomes unaffordable and data unavailable. Web 2.0 tools do not have formal models that allow the creation of complex systems managing large amounts of data. For this reason it is possible to improve any of the six components of Enterprise 2.0 technologies described by McAfee [4] by the introduction of semantic analysis of the content:

- Search: search should not be limited to documents. It should also consider people, their interests and their knowledge.
- Links: links between documents are important but links between contents could be more powerful. Identifying semantically related content helps users in the creation and discovery of contents. Semantic links between people sharing the same knowledge should also be considered.
- Authoring: any Enterprise 2.0 platform should rely on solid mechanisms for an organic growth of the contents and tools for an easy and user-friendly content creation have to be provided. Semantic tools for automatic information discovery during the process of creation help in this task.
- Tags: solutions like folksonomies (folk's taxonomies), collaborative tagging and social tagging are adopted for collaborative categorization of contents. In this scenario we have to face the problem of scalability and interoperability [8]: making users free to use any keyword is very powerful but this approach does not consider the natural semantic relations between the tags.
- Extensions: Recommendations systems are key when the user faces a large amount of contents. Automatic recommendations ("Users like you also like") could benefit from semantic analysis of contents. Through the semantic analysis of documents created by a specific user it is possible to recommend semantically similar documents.
- Signals: Users can easily feel overwhelmed by the large amount of information. It is necessary to introduce technology to signal users when new content of interest appears. RSS allow users to subscribe to sources. Semantic technology allows the creation of automatic alerts for new interesting content based on semantic analysis.

Semantic Web can contribute introducing computer-readable representations for simple fragments of meaning. As we will see in the following paragraphs, an ontology-based analysis of a plain text provides a semantic contextualization of the contents, supports tasks such as finding semantic distance between contents and helps in creating relations between people with shared knowledge and interests. Both Web 2.0 principles and semantic technologies can be naturally applied to the innovation process for the following reasons:

- Communication is essential for knowledge creation: semantic technologies facilitate information exchange;
- Knowledge is created and shared within complex processes;
- Knowledge can still be unavailable to some members of the community due to the dimension of the organization;
- Innovation needs communities composed of heterogeneous groups of members with different skills and interests: Web 2.0 is based on cooperative work;
- Information flows are associated with knowledge creation. This flow contains information about expertise, relationships among members etc. If a specific tool does not handle this data, it is easily lost: semantic technologies support access to unstructured data.

Thanks to the adoption of these technologies an innovation culture is created in the firm; while the support of collaborative work, talent management and proposal evaluation provides the enhancement of efficacy and efficiency of the innovation process.

2.1 Actors and Tools in the Corporate Scenario

Enterprise 2.0 technologies have the potential to usher in a new era by making both the practices of knowledge work and its output more visible. Lee [9] states that to increase the participation in online communities is necessary to create a common space for collaboration, to have a common ground, to make members of the community aware of the presence and work of other actors, and introduce interaction enablers and mechanisms. Therefore, a Web 2.0 shared platform to perform interaction has to be introduced: blogs, forums and wikis are the best environment to support the exchange of ideas, create new knowledge and suggest solutions and improvements for the organization. Here the “content” is stored. In addition, a repository of knowledge, such as an ontology, is required. The ontology provides a formal description of the domain in which community members interact. Terms in the ontology represent the bridge between people and contents: the detection of semantic relationships between contents and people through the identification of concepts is the core of collaboration support.

Moreover, the adoption of a reward system is key to involve people in the innovation process. Money is not the sole motivating factor. There may be other factors such as prestige and ego. A company could collaborate in another firm’s innovation process as a marketing strategy, in order to gain public recognition as an “innovative partner”. Technology has to support the innovation process in this aspect as well, helping decision makers in the enterprise to evaluate the ideas and to reward the members of the community.

2.2 Innovation Process Stages and Idea Lifecycle

The innovation process could be divided in six stages: creation, filtering, decision making, acknowledgment, implementation and exploitation.

During the creation phase a member of the community creates a new idea. The filtering is carried out by the collaborative work of the community (editing, voting and

commenting on existing ideas). A group of experts should then evaluate the idea before the innovation managers decide whether to implement the idea or to hibernate it. For the ideas which are going to be implemented the author(s) receive social acknowledgment and, depending on the organization, financial reward. Our analysis focuses on the three first stages in which a technological framework is needed.

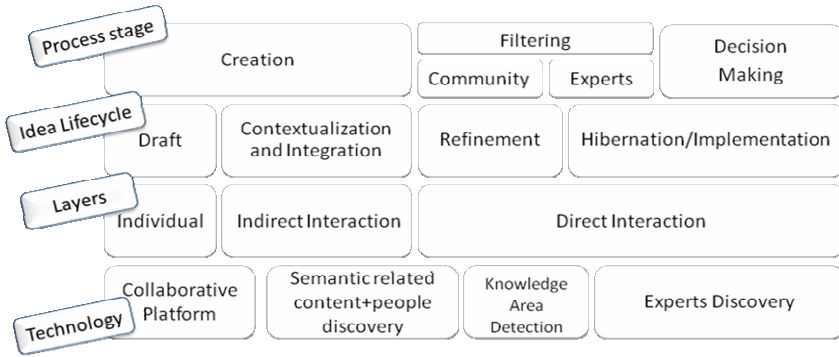


Fig. 2. Innovation stages, idea lifecycle and the corresponding interaction layers

The “idea lifecycle” is the process the ideas go through during the innovation stages: a draft is uploaded into a collaboration platform (such as a wiki, blog or a forum could be) where it can be related and compared to other ideas (contextualization), modified, voted and commented on (refinement) and finally evaluated (hibernation or implementation). For each of these stages it is possible to identify different types of interaction we can organize in layers:

- Individual Layer: a user or a user group sketches a draft not integrated or related to other proposals. There is no interaction with the rest of the community.
- Indirect Interaction Layer: the idea is contextualized and introduced into the interaction platform. The platform can support the authors to discover similar proposals and integrate their idea with others, so that they interact with the community in an indirect way through the ideas previously proposed.
- Direct Interaction Layer: collaboration between the members of the community is actively supported. The authors may consult other members to refine their proposal, and community members are invited to comment and modify the idea. People with common interests are put in touch for creating new proposals. If the idea matures enough it is evaluated. Evaluation is managed from the platform and performed by a group of experts selected from the community to assess specific aspects, such as technical or commercial viability. The outcome of this process will help decision makers in the R&D department to select the ideas to be finally implemented.

Each layer requires a technological framework in order to support and improve collaboration between the members of the community, the evaluation and the decision.

2.3 The Individual Layer

Adopting a market metaphor, people are creators and consumers of contents which can be stored in a semi-structured workspace, such as a wiki. Wiki pages can be easily created and updated and foster collaboration in document creation. Contents are characterized by the concepts they contain. The first two basic functionalities we identify for collaboration support consist in finding (i) “what” a specific content is about and (ii) “what” a member of the community consumes and generates. In this way, concepts are associated with contents and concepts with people. Creators know what they write; consumers are interested about what they read.

In order to perform this task, manual tagging could be used (defining this way a folksonomy) but semantic tools for automatic concept recognition grant a more formal identification of the content of the document: the user receives suggestions about the identified concepts in the text and is free to choose which ones better describe the content. Concept detection in contents and about people characterizes the individual layer.

2.4 The Indirect Interaction Layer

By “indirect interaction” we mean the creation and consumption of content, which does not require direct collaboration. In this context, members of the community proactively decide to participate in the community life and the system only supports their tasks. The functionalities we identify for this layer are: (i) search and detection of concept-related contents, (ii) discovery of personal interest and (iii) discovery of personal knowledge.

The information about the “meaning” of contents leads directly to the discovery of “concept-related” contents. Once again, the knowledge stored in the ontology allows a semantic comparison between contents. A simple keywords comparison between two texts would provide a measure given by the number of common terms. Adopting a semantic approach, not just the appearance of the same terms is considered, but also synonymy and semantic distance in the ontology is taken into account in order to establish the degree of similarity. For example, a text about “financial products” and another about “hedge funds” are somehow related. The distance given by the hierarchy and the relations between terms of the ontology reveals how similar the two concepts are. The higher the distance, the lower the degree of similarity is. We will describe in the next paragraph how this distance is calculated.

Detection of related content is necessary whether new content is created or consumed. The process of generating new contents takes into account ideas already introduced by other members of the community in order to avoid duplicated information and to help in the edition of new contents.

Concepts are grouped in “knowledge areas” in the ontology and we can say that people and contents are related not just to single terms but have a more wide relation to knowledge areas of the corporation. Knowledge areas are sets of concepts of the ontology, and can have intersections and be subsets of each other. Knowing which concepts a person handles and knowing which concept belongs to which knowledge area it is possible to calculate how a person (or content) is related to an area. Information retrieval algorithms can be applied for determining which knowledge areas are involved in a text.

Concepts related to a person reveal important information about the skills, knowledge and interests of people and help in talent management. A content consumer is making explicit her interest about specific concepts or specific knowledge areas. A content creator demonstrates her knowledge and skills whenever she writes a new proposal. The knowledge is given by the set of terms this person writes, while interests are the set of words a person reads. The definition of knowledge areas is part of the modeling of the domain and is included in the ontology.

2.5 The Direct Interaction Layer

It is not always possible to rely on the proactive impulse of the members of the community to collaborate without an external influence. Semantic technologies can be used to implement functionalities to involve passive members in the cooperative community life. Content creators are asked to rate recently generated content covering topics similar to the ones they are semantically related to.

Furthermore, the creation of content should be a collaborative act: while the user introduces text in the interaction platform, not just related ideas are shown in real time but also people with related knowledge. Handling knowledge (and interests) and sets of words allow the comparison of people and ideas as we can compare plain texts: tools to calculate semantic distance used for discovering similar ideas are still valid. By tracking user content consumption we obtain the set of words, which represents her interests. This data is used to inform a person about new interesting content.

The number of co-authorships, revisions, comments to each proposal provides an impact index of an idea and a map of the relations between members of the community.

A feedback system is included in the interaction platform allowing members of the community to rate ideas. This tool becomes very important if the discovery of new experts of specific knowledge areas is to be achieved: a person who produces valuable content about a topic (a knowledge area) is a potential new expert in the area. In conclusion, we identified two main functionalities in this layer: (i) content rating and broadcasting and (ii) updating of the community members' interactions map (detection of people with related interests or knowledge, discovery of new potential experts, etc).

3 Implementation Issues

In this section we give some details about the implementation of the three main components of the model: the interaction platform, the ontology and the semantic engine for text comparison.

The interaction platform is a web application designed following the Web 2.0 principles of participation and usability. Every employee of the three different enterprises where the system has been adopted has access to the company innovation platform. Challenges for innovation about a specific topic are proposed by the R&D department. Employees can participate in the challenge and compete for a prize for the best idea or proactively propose an idea not related to any active challenge. An innovation committee is in charge of taking the final decision about which ideas to implement (and reward) or hibernate at a given deadline. An idea is similar to a wiki entry which any member of the community is free to edit. If the blog philosophy is adopted, only

the authors can edit the idea. Versions, changes and information about the users involved in the definition of the idea are kept for the evaluation of the contribution of every author in case a reward is given. Users can also comment or vote for ideas. Interactions between community members (co-authoring, voting, or writing a comment) are stored in order to draw an interaction map for each user in her personal space where other information such as related people or new interesting ideas identified by the system is presented.

The semantic functionalities are implemented in a three layered architecture as shown in Figure 3: ontology and ontology access is the first layer, keyword to ontology entity mapping is the second and the last layer is semantic indexing and search.

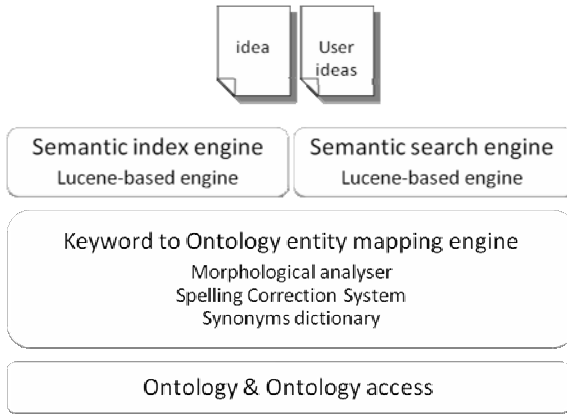


Fig. 3. The semantic architecture

For the three different companies, three ontologies, each one modeling the specific business field, have been implemented in RDF/RDFS. Knowledge engineers and domain experts worked together to define concepts and relations in the ontologies. General aspects, such as “product”, “service”, “process”, “customer” are common terms for the three ontologies, while specific terms have been added for each case. Ontologies are accessed through the Sesame RDF framework⁴.

An engine to map keywords to ontology entities has been implemented in order to detect which terms (if any) in the text of an idea are present in the ontology. For this task we consider: morphological variations, orthographical errors and synonyms (for the terms defined in the ontology). Synonyms are manually defined by knowledge engineer and domain experts as well.

The indexes and the search engine are based on Lucene⁵. Three indexes have been created: ideas index, user interests index (using the text of the ideas the user reads) and user knowledge interests index (using the text of the ideas the user writes). Each index contains terms tokenized using blank space for word delimitation and ontology terms as single tokens (e.g. if the text contains “credit card” and this is a term of the

⁴ <http://www.openrdf.org>

⁵ <http://lucene.apache.org/java/docs/>

ontology, “credit”, “card” and “credit card” are added as tokens to the index). When we look for related ideas to a given one the following tasks are executed:

- extraction of the text of the idea for using it as a query string;
- morphological analysis;
- ontology terms identification (considering synonyms);
- query expansion exploiting ontological relations. If a synonym of an ontology term is detected, the ontology term is added to the query. If a term corresponding to an ontology class is found, subclasses and instances labels are used to expand the query. If an instance label is identified, the corresponding class name and sibling instance labels are added to the query. Different boosts are given to the terms used for each different query expansion.

The same tasks are performed for searches of related people. For expert detection, semantic search results are filtered with statistical results about successful ideas.

4 Evaluation Issues

In order to evaluate the impact of the introduction of this new paradigm and technologies new metrics should be considered. The number of new products and the number of registered patents gives the measure of the success of the R&D department in a closed model. In an open innovation scenario, time to market and the number of successful proposals (or patents in use) has to be evaluated. Our experience in the three cases of studies has not yet allowed a deep analysis of the financial outcome and detailed statistics cannot be given. Nevertheless the adoption of the model did cause an increase in the number of proposals generated from many different departments of the firms. It is also important to highlight the high collaboration rate in the proposals definition: the Web 2.0 environment helped in establishing an innovation culture in the firms, while the semantic technologies helped not just in fostering interaction for the creation of new ideas, but also in supporting the decision process. The higher number of proposals is balanced by the collaborative filtering of the community (through the rating system and experts evaluation) and only valuable ideas reach the last stage of decision making. The impact index pointed out the most debated proposals offering a pulse of the interests of the community. Talent management profited from all the information given about created and consumed contents by every member of the community. R&D experts in the firms perceived a more efficient and effective innovation process and some of the information obtained thanks to the use of the system was shared with Knowledge Management and Human Resources units.

In two cases (in the bank and in the energy firm), departments other than R&D are studying the introduction of this model for improving collaboration and knowledge exchange in their business unit.

5 Conclusions and Future Work

This paper introduces real experiences of the combination of collaborative technology (Web 2.0 like) and knowledge technology (Semantic Web like) to develop successful

solutions for supporting knowledge intensive business processes. This paper described the actors and tools involved in a three-layered interaction model for open innovation in which Web 2.0 and semantic technologies support knowledge management, cooperation and evaluation of the proposals. We have been able to measure the impact of these technologies in the innovation process at its first stages: collaboration increased and the evaluation process is perceived as more effective (according to the theory of the wisdom of crowds). Our future work includes the refinement of the model, a deeper analysis of the outcome after the adoption of this model and the definition of more exact metrics.

References

1. O'Reilly, T.: What is Web 2.0: Design Patterns and Business Models for the next generation of software (2005), <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>
2. Battelle, J., O'Reilly, T.: Opening Welcome: The State of the Internet Industry. In: Opening Talk at the Web 2.0 Conference, San Francisco U.S.A. (2004)
3. Fensel, D., Wahlster, W., Lieberman, H., Hendler, J.: Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. MIT Press, Cambridge (2002)
4. McAfee, A.P.: Enterprise 2.0: The Dawn of Emergent Collaboration. MIT Sloan Management Review 47(3), 20–28 (2006)
5. Chesbrough, H., Vanhaverbeke, W., West, J.: Open Innovation: Researching a New Paradigm. Oxford University Press, Oxford (2006)
6. Platt, M.: Web 2.0 in the Enterprise. The Architecture Journal (2007)
7. Gloor, P.A.: Swarm Creativity: Competitive Advantage through Collaborative Innovation Networks. Oxford University Press, Oxford (2006)
8. Greaves, M.: The Relationship Between Web 2.0 And the Semantic Web. In: ESTC 2007, Wien, Austria (2007)
9. Lee, A., Danis, C., Miller, T., Jung, Y.: Fostering social interaction in online spaces. In: International Conference on Human-Computer Interaction, pp. 59–66. IOS Press, Amsterdam (2001)

Algorithmic Decision of Syllogisms*

Bora İ. Kumova and Hüseyin Çakır

İzmir Institute of Technology, Department of Computer Engineering, 35430 Turkey
{borakumova,huseyincakir}@iyte.edu.tr

Abstract. A syllogism, also known as a rule of inference, is a formal logical scheme used to draw a conclusion from a set of premises. In a categorical syllogisms, every premise and conclusion is given in form a of quantified relationship between two objects. The syllogistic system consists of systematically combined premises and conclusions to so called figures and moods. The syllogistic system is a theory for reasoning, developed by Aristotle, who is known as one of the most important contributors of the western thought and logic. Since Aristotle, philosophers and sociologists have successfully modelled human thought and reasoning with syllogistic structures. However, a major lack was that the mathematical properties of the whole syllogistic system could not be fully revealed by now. To be able to calculate any syllogistic property exactly, by using a single algorithm, could indeed facilitate modelling possibly any sort of consistent, inconsistent or approximate human reasoning. In this paper we present such an algorithm.

Keywords: Syllogistic reasoning, fallacies, automated reasoning, approximate reasoning, human-machine interaction.

1 Introduction

The first studies on syllogisms were pursued in the field of right thinking by the philosopher Aristotle [1]. His syllogisms provide patterns for argument structures that always yield conclusions, for given premises. Some syllogisms are always valid for given valid premises, in certain environments. Most of the syllogisms however, are always invalid, even for valid premises and whatever environment is given. This suggests that structurally valid syllogisms may yield invalid conclusions in different environments.

Given two relationships between the quantified objects P, M and S, a syllogism allows deducing a quantified transitive object relationship between S and P. Depending on alternative placements of the objects within the premises, 4 basic types of syllogistic figures are possible. Aristotle had specified the first three figures. The 4. figure was discovered in the middle age. In the middle of the 19th century, experimental studies about validating invalid syllogisms were pursued. For instance, Reduction of a syllogism, by changing an imperfect mood into a perfect one [13]. Conversion of a mood, by transposing the terms, and thus drawing another proposition from it of the same quality [11], [10].

* This research was partially funded by the grant project 2009-İYTE-BAP-11.

Although shortly thereafter syllogism were superseded by propositional logic [7], they are still matter of research. For instance philosophical studies have confirmed that syllogistic reasoning does model human reasoning with quantified object relationships [2]. For instance in psychology, studies have compared five experimental studies that used the full set of 256 syllogisms [4], [12] about different subjects. Two settings about choosing from a list of possible conclusions for given two premisses [5], [6], two settings about specifying possible conclusions for given premisses [8], and one setting about decide whether a given argument was valid or not [9]. It has been found that the results of these experiments were very similar and that differences in design appear to have had little effect on how human evaluate syllogisms [4]. These empirically obtained truth values for the 256 moods are mostly close to their mathematical truth ratios that we calculate with our algorithmic approach.

Although the truth values of all 256 moods have been analysed empirically, mostly only logically correct syllogisms are used for reasoning or modus ponens and modus tollens, which are generalisations of syllogisms [14]. Uncertain application environments, such as human-machine interaction, require adaptation capabilities and approximate reasoning [16] to be able to reason with various sorts of uncertainties. For instance, we know that human may reason purposefully fallacious, aiming at deception or trickery. Doing so, a speaker may intent to encourage a listener to agree or disagree with the speaker's opinions. For instance, an argument may appeal to patriotism, family or may exploit an intellectual weakness of the listener. We are motivated by the idea for constructing a fuzzy syllogistic system of possibilistic arguments for calculating the truth ratios of illogical arguments and approximately reason with them.

This paper presents an algorithm for deciding syllogistic cases, for algorithmically calculating syllogistic reasoning and an application to automated reasoning. Firstly, categorical syllogisms are discussed briefly. Thereafter an arithmetic representation for syllogistic cases is presented, followed by an approach for algorithmically deciding syllogisms and a possible application for recognising fallacies and reasoning with them.

2 Categorical Syllogisms

A categorical syllogism can be defined as a logical argument that is composed of two logical propositions for deducing a logical conclusion, where the propositions and the conclusion each consist of a quantified relationship between two objects.


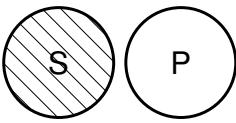
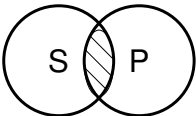


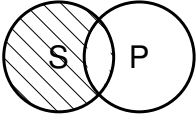
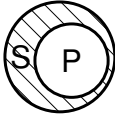
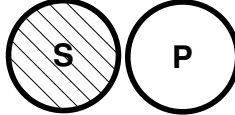
2.1 Syllogistic Propositions

A syllogistic proposition or synonymously categorical proposition specifies a quantified relationship between two objects. We shall denote such relationships with the operator ψ . Four different types are distinguished $\psi \in \{A, E, I, O\}$ (Table 1):

- A is universal affirmative: All S are P
- E is universal negative: All S are not P
- I is particular affirmative: Some S are P
- O is particular negative: Some S are not P

One can observe that the proposition I has three cases (a), (b), (c) and O has (a), (b), (c). The cases I (c) and O (c) are controversial in the literature. Some do not consider them as valid [3] and some do [15]. Since case I (c) is equivalent to proposition A, A becomes a special case of I. Similarly, since case O (c) is equivalent to proposition E, E becomes a special case of O. At this point we need to note however that exactly these cases complement the homomorphic mapping between syllogistic cases and the set-theoretic relationships of three sets. This is discussed below.

Table 1. Syllogistic propositions consist of quantified object relationships

| Operator ψ | Proposition Φ | Set-Theoretic Representation of Logical Cases |
|-----------------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A | All S are P |  |
| E | All S are not P |  |
| I | Some S are P |    (a) (b) (c) |
| O | Some S are not P |    (a) (b) (c) |

2.2 Syllogistic Figures

A syllogism consists of the three propositions major premise, minor premise and conclusion. The first proposition consist of a quantified relationship between the objects M and P, the second proposition of S and M, the conclusion of S and P (Table 2).

Since the proposition operator ψ may have 4 values, 64 syllogistic moods are possible for every figure and 256 moods for all 4 figures in total. For instance, AAA-1 constitutes the mood MAP, SAM - SAP in figure 1. The mnemonic name of this mood is Barbara, which comes from syllogistic studies in medieval schools. Mnemonic names were given to each of the in total 24 valid moods, out of the 256, for easier memorising them [3].

Table 2. Syllogistic figures

| Figure Name | I | II | III | IV |
|---------------|-----------|-----------|-----------|-----------|
| Major Premise | $M\psi P$ | $P\psi M$ | $M\psi P$ | $P\psi M$ |
| Minor Premise | $S\psi M$ | $S\psi M$ | $M\psi S$ | $M\psi S$ |
| Conclusion | $S\psi P$ | $S\psi P$ | $S\psi P$ | $S\psi P$ |

We shall denote a propositional statement with Φ_i , in order to distinguish between possibly equal propositional operators of the three statements of a particular mood, where $i \in \{1, 2, 3\}$.

A further consequence of including the above mentioned cases I (c) and O (c) in our algorithmic approach is that the number of valid moods increases with AAO-4 from 24 to 25. Since no mnemonic name was given to this mood in the literature by now, name it herewith "anasoy".

3 Algorithmic Representation

In the following our approach for algorithmically deciding any given syllogistic mood is presented. Algorithmically analysing all 2624 truth cases of the 256 moods enables us to calculate all mathematical truth values of all moods, sort the moods according their truth values and define a fuzzy syllogistic system of possibilistic arguments.

3.1 Set-Theoretical Analysis

For three symmetrically intersecting sets there are in total 7 possible sub-sets in a Venn diagram (Fig 1). If symmetric set relationships are relaxed and the three sets are named, for instance with the syllogistic terms P, M and S, then 41 set relationships are possible. These 41 relationships are distinct, but re-occur in the 256 moods as basic syllogistic cases. The 7 sub-sets in case of symmetric relationships and the 41 distinct set relationships in case of relaxed symmetry are fundamental for the design of an algorithmic decision of syllogistic moods.

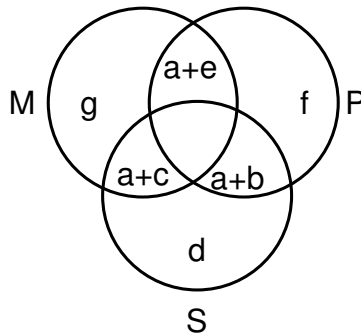


Fig. 1. Mapping the sub-sets of the symmetrically intersecting sets P, M and S onto arithmetic relations

We have pointed out earlier that, including the cases I (c) and O (c) of the syllogistic propositions I and O, is required by the algorithm to calculate correctly. Without these cases, the algorithm presented below, cannot decide some cases of some moods or cannot find valid moods at all. For instance, as valid moods in figure I, only AAA, AAI, AII and EAE can be found by the algorithm, although EAO and EIO are also true. If the algorithm considers the cases I (c) and O (c), then all 6 valid moods of figure I are found. The reason for that is that the syllogistic propositions are basically a symmetric sub-set of the in total 12 distinct set relationships between two named sets. Therefore the cases I (c) and O (c) are required to complement the symmetric relationships between the syllogistic propositions.

3.2 Arithmetic Representation

Based on these 7 sub-sets, we define 9 distinct relationships between the three sets P, M and S (Table 3). These 9 relationships are mapped homomorphically onto the 9 arithmetic relations, denoted with $\delta_1, \dots, \delta_9$. For instance $P \cap M$ is mapped onto $\delta_1 = a + e$ and $P - M$ is mapped onto $\delta_4 = f + b$. These relationships can be verified visually in the Venn diagram (Fig 1).

One can observe that the symmetric relationship between the three sets (Fig 1) is preserved in the homomorphically mapped arithmetic relations (Table 3).

Table 3. Homomorphism between the 9 basic syllogistic cases and 9 arithmetic relations

| Sub-Set Number | δ_1 | δ_2 | δ_3 | δ_4 | δ_5 | δ_6 | δ_7 | δ_8 | δ_9 |
|---------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| Arithmetic Relation | a+e | a+c | a+b | f+b | f+e | g+c | g+e | d+b | d+c |
| Syllogistic Case | $P \cap M$ | $M \cap S$ | $S \cap P$ | $P - M$ | $P - S$ | $M - P$ | $M - S$ | $S - M$ | $S - P$ |

The above homomorphism represents the essential data structure of the algorithm for deciding syllogistic moods.

3.3 Algorithmic Decision

The pseudo code of the algorithm for determining the true and false cases of a given moods is based on selecting the possible set relationships for that mood, out of all 41 possible set relationships.

```

DETERMINE mood
  READ figure number {1,2,3,4}
  READ with 3 proposition ids {A,E,I,O}
GENERATE 41 possible set combinations with 9
  relationships into an array
setCombi[41,9]={{1,1,1,1,1,1,1,1,1}, ... ,
  {0,1,0,0,1,1,1,1,1}}
VALIDATE every proposition with either validateAllAre,
  validateAllAreNot, validateSomeAreNot or
  validateSomeAre

```

```

DISPLAY valid and invalid cases of the mood
VALIDATE mood
validateAllAre(x,y) //all M are P
    if(x=='M' && y=='P')
        CHECK the sets suitable for this mood in setCombi
        if  $\delta_1=1$  and  $\delta_2=0$  then add this situation as valid
            if(setCombi[i][0]==1 && setCombi[i][1]==0)
//similar for validateAllAreNot(), validateSomeAre(),
    validateSomeAreNot()

```

3.4 Statistics about the Syllogistic System

The introduced algorithm enables revealing various interesting statistics about the structural properties of the syllogistic system. Some of them are presented now.

Since our objective is to utilise the full set of all 256 moods as a fuzzy syllogistic system of possibilistic arguments, we have first calculated the truth values for every mood in form of a truth ration between its true and false cases, so that the truth ratio becomes a real number, normalised within $[0, 1]$. Thereafter we have sorted all moods in ascending order of their truth ratio (Fig 2). Note the symmetric distribution of the moods according their truth values. 25 moods have a ratio of 0 (false) and 25 have ratio 1 (true). 100 moods have a ratio between 0 and 0.5 and 100 have between 0.5 and 1. 6 moods have a ratio of exactly 0.5.

Every mood has 0 to 21 true and 0 to 21 false cases, which is a real sub-set of the 41 distinct cases. The total number of true or false cases varies from one mood to another, from 1 to 24 cases. For instance, mood AAA-1 has only 1 true and 0 false cases, whereas mood OIA-1 has 3 true and 21 false cases. Hence the truth ratio of AAA-1 is 1 and that of OIA- is $3/21=1/7$. The algorithm calculates 2624 syllogistic cases in total, since all cases of the 256 moods map the 41 distinct cases multiple times. Interesting is also that for any given figure the total number of all true cases is equal to all false cases, ie 328 true and 328 false cases. Thus we get for all 4 syllogistic figures the total number of $4 \times 2 \times 328 = 2624$ cases. More statistical details will be discussed in a separate work.

3.5 Fuzzy Syllogistic System

Based on the structural properties of the syllogistic system, we elaborate now a fuzzified syllogistic system.

One can see (Fig 2) that every syllogistic case is now associated with an exact truth ration. We utilise the symmetric distribution of the truth ratios, for defining the membership function $FuzzySyllogisticMood(x)$ with a possibility distribution that is similarly symmetric (Fig 2). The linguistic variables were adopted from a meta membership function for a possibilistic distribution of the concept likelihood [17]. The complete list with the names of all 256 moods is appended (Table A1).

As we have mentioned earlier, the algorithmically calculated truth ratios of the 256 moods (Fig 2) mostly comply with those empirically obtained truth ratios in psychological studies [4]. Hence the suggested possibilistic interpretation should reflect an approximately correct model of the syllogistic system.

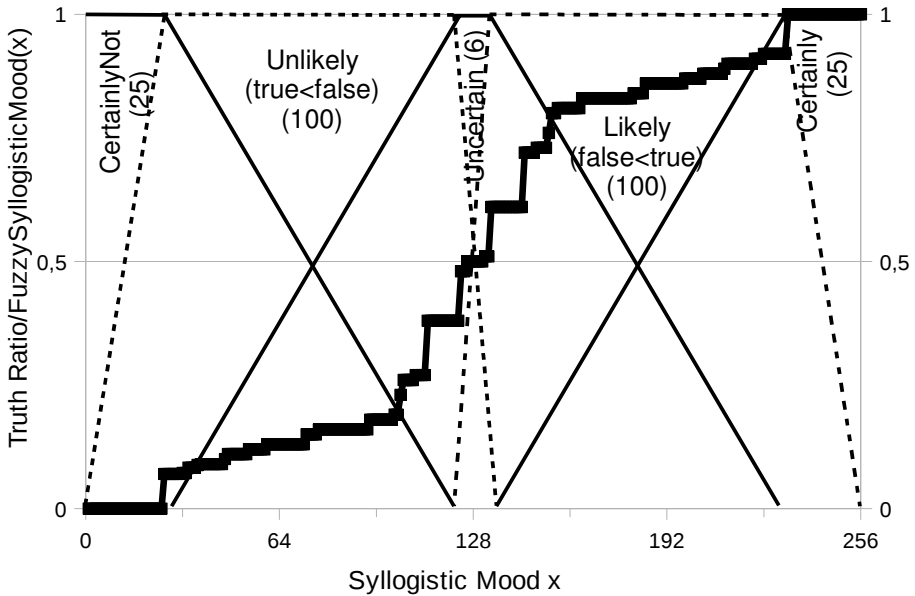


Fig. 2. 256 syllogistic moods sorted in ascending order of their truth ratio true/false, if number of truth cases of a mood is true<>false and false/true ratio, if false<true. Definition of the possibility distribution FuzzySyllogisticMood(x) with the linguistic variables CertainlyNot, Unlikely, Uncertain, Likely, Certainly and their cardinalities 25, 100, 6, 100, 25, respectively.

4 Recognising Fallacies and Fuzzy Syllogistic Reasoning

In logic a fallacy is a misconception resulting from incorrect reasoning in argumentation. 7 syllogistic fallacies are known in the literature:

- Equivocation fallacy or fallacy of necessity: Unwarranted necessity is placed in the conclusion, by ignoring other possible solutions.
- Fallacy of undistributed middle: Middle term must be distributed in at least one premiss.
- Illicit major/minor: No term can be distributed in the conclusion, which is not distributed in the premisses.
- Fallacy of exclusive premisses: Two negative premisses.
- Affirmative conclusion from negative premiss: Positive conclusion, but at least one negative premiss.
- Existential fallacy: Two universal premisses, but particular conclusion.

These fallacies comply exactly with the 7 rules for eliminating invalid moods, which were discovered already by Aristotle [1].

Our objective is to use the whole set of 256 syllogistic moods as one system of possibilistic arguments for recognising fallacies and reasoning with them. For that purpose, we specify the following steps:

- Calculate all truth cases and truth ratio of a given mood.
- Try to recognise fallacies by
 - + identifying false or true possibilities: reduction of A to I or E to O, respectively
 - + generalising true or false possibilities: generalisation of I to A or O to E.
- Try to map the initial mood to a mood with a truth ratio closer to 1.
- Approximately reason with the truth ratios.

We will now discuss these steps experimentally on the following example (Fig 3). Firstly, we calculate the 3 true (Fig 4) and 3 false (Fig 5) cases of mood AIA-1 and its truth ratio of 0.5. Secondly, we identify following fallacies:

- $\neg\Phi_1(A)$: Simply not all stories in The Child's Magic Horn are sad. The truth is that only some stories in The Child's Magic Horn are sad $\Phi_1(I)$.
- $\neg\Phi_3(A)$: Not all stories I cry at are stories in The Child's Magic Horn, because I will possibly cry at some other stories as well. The truth is that only some of all the stories I cry at are stories in The Child's Magic Horn $\Phi_3(I)$.

Thirdly, based on the identified fallacies and reductions to $\Phi_1(I)$ and $\Phi_3(I)$, we can easily calculate the mood III-1 to be "more true" for the given sample propositions. In dead, mood III-1 has with 4 false/19 true cases = 0.73, a better truth ratio.

In the last step, we may use the truth ration of the mood for fuzzy syllogistic reasoning as a model for approximate reasoning with quantified propositions.

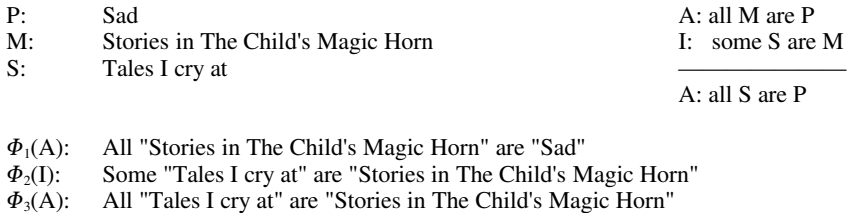


Fig. 3. Sample syllogistic inference with the mood AIA1

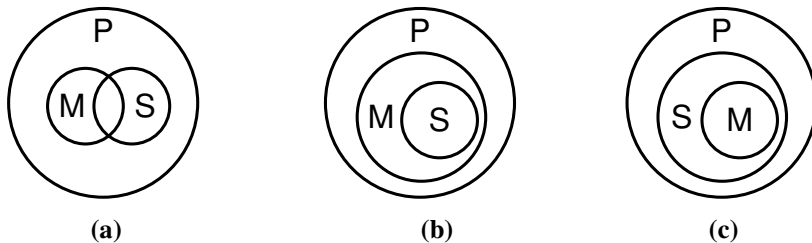


Fig. 4. True syllogistic cases of the mood AIA1

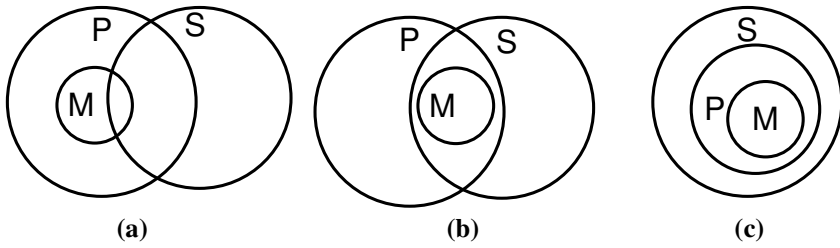


Fig. 5. False syllogistic cases of the mood AIAI

5 Conclusion

We have presented an algorithmic approach for analysing the syllogistic system. The algorithm facilitates structurally analysing the syllogistic moods and reveals interesting statistics about the truth cases of the moods. First experimental results show that the syllogistic system is inherently symmetric.

With the membership function $\text{FuzzySyllogisticMood}(x)$ we have proposed a fuzzy syllogistic system of possibilistic arguments and its possible application for recognising fallacies and fuzzy syllogistic reasoning.

Future work shall aim at systematically revealing all significant statistical properties of the syllogistic system, by using the algorithm. We believe that this approach may prove a practical approach for reasoning with inductively learned knowledge, where P, M, S object relationships can be learned inductively and the "most true" mood can be calculated automatically for those relationships. That shall be our future work, along with examples including recognising intentional or unintentional fallacies, with the objective to facilitate automated human-machine interaction.

References

- [1] Aristotle: *The Works of Aristotle*, vol. 1. Oxford University Press, Oxford (1937)
- [2] Geurts, B.: *Reasoning with quantifiers*, Department of Philosophy. University of Nijmegen (2002)
- [3] Brennan, J.G.: *A Handbook of Logic*. Brennan Press (2007)
- [4] Chater, N., Oaksford, M.: The probability heuristics model of syllogistic reasoning. *Cognitive Psychology* 38, 191–258 (1999)
- [5] Dickstein, L.S.: The effect of figure on syllogistic reasoning. *Memory and Cognition* 6, 76–83 (1978)
- [6] Dickstein, L.S.: Conversion and possibility in syllogistic reasoning. *Bulletin of the Psychonomic Society* 18, 229–232 (1981)
- [7] Frege, L.G.F.: *Begriffsschrift, eine der Arithmetischen Nachgebildete Formalsprache des Reinen Denkens*. Verlag von Louis Nebert (1879)
- [8] Johnson-Laird, P.N., Steedman, M.: The psychology of syllogisms. *Cognitive Psychology* 10, 64–99 (1978)
- [9] Johnson-Laird, P.N., Bara, B.G.: Syllogistic inference. *Cognition* 16, 1–61 (1984)

- [10] Leechman, J.: Study of Reasoning, ch. VIII, pp. 89–100. Irregular Syllogisms (1864)
- [11] Morell, J.D.: Hand-Book of Logic. Longman (1857)
- [12] Oaksford, M., Chater, N.: The probabilistic approach to human reasoning. Trends in Cognitive Sciences 5, 349–357 (2001)
- [13] Parker, S.E.: Logic or the Art of Reasoning Simplified. Harvard College Library (1837)
- [14] Russell, S., Norvig, P.: Artificial Intelligence - A Modern Approach. Prentice-Hall, Englewood Cliffs (2009)
- [15] Wille, R.: Contextual Logic and Aristotle's Syllogistic. Springer, Heidelberg (2005)
- [16] Zadeh, L.A.: Fuzzy Logic and Approximate Reasoning. Syntheses 30, 407–428 (1975)
- [17] Zadeh, L.A., Bellman, R.E.: Local and fuzzy logics. In: Dunn, J.M., Epstein, G. (eds.) Modern Uses of Multiple-Valued Logic. Reidel, Dordrecht (1977)

Appendix A: Truth Degree of Syllogistic Moods

The table (Table A1) shows the 256 moods in 5 categories with truth ratio normalised in [0,1]. False, undecided and true moods are not sorted. Unlikely and Likely moods are sorted in ascending order of their truth ratio. The table also shows the possibility distribution of the membership function $FuzzySyllogisticMood(x)$, with $x \in \{CertainlyNot, Unlikely, Uncertain, Likely, Certainly\}$, defined over the truth ratios of the moods.

Table A1. Possibility distribution $FuzzySyllogisticMood(x)$ over the Syllogistic moods in increasing order of truth ratio of the moods

| Linguistic Variables | Sum | Moods |
|-------------------------------------------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CertainlyNot; false; ratio=0 | 25 | AAE-1, AAO-1, AIE-1, EAA-1, EAI-1, EIA-1, AEA-2, AEI-2, AOA-2, EAA-2, EAI-2, EIA-2, AAE-3, AIE-3, EAA-3, EIA-3, IAE-3, OAA-3, AAA-4, AAE-4, AEA-4, AEI-4, EAA-4, EIA-4, IAE-4 |
| Unlikely; rather false; $0 < ratio < 0.5$ | 100 | EIE-1, IEE-1, EIE-2, IEE-2, EIE-3, IEE-3, EIE-4, IEE-4, AOE-2, OAA-2, OAE-2, AOA-1, IAA-1, OAE-1, OEE-1, IAA-2, EOE-3, OEE-3, AOE-4, EOE-4, OOE-3, AEA-1, AEE-1, AAA-3, AEA-3, AEE-3, EAE-3, EAE-4, EOE-1, EOE-2, OEA-2, OEE-2, OEA-4, OEE-4, OIE-1, OOE-1, OOA-4, OOE-4, IOA-3, IOE-3, OIE-3, IOA-4, IOE-4, IEA-1, IEA-2, IEA-3, IEA-4, IIA-1, IIA-2, IIA-3, IIA-4, IAE-1, OAA-1, OEA-1, AIE-2, IAE-2, OEA-3, AIE-4, AAA-2, AAE-2, EAA-1, EEE-1, EEA-2, EEE-2, EEA-3, EEE-3, EEA-4, EEE-4, IOA-1, IOE-1, IOA-2, IOE-2, OIA-2, OIE-2, OIA-4, OIE-4, OOA-2, OOE-2, OOA-3, IIE-1, IIE-2, IIE-3, IIE-4, AOE-3, IAA-3, OAE-3, IAA-4, OOA-1, OIA-1, OIA-3, AOE-1, AIA-2, EOA-3, AIA-4, AOA-4, EOA-4, OAA-4, OAE-4, EOA-1, EOA-2 |
| Uncertain; undecided; ratio=0.5 | 6 | AIA-1, AIO-1, AIA-3, AIO-3, AOA-3, AOO-3 |
| Likely; rather true; $0.5 < ratio < 1.0$ | 100 | EOO-1, EOO-2, OIO-1, OOO-1, OIO-3, AIO-2, EOO-3, AIO-4, AOI-1, AOO-4, EOO-4, OAI-4, OAO-4, IAO-3, IAO-4, OAI-3, AOI-3, III-1, III-2, III-3, III-4, OOO-3, OOI-2, OOO-2, IOI-1, IOO-1, OII-2, OIO-2, IOI-2, IOO-2, OII-4, OIO-4, IAI-1, OAO-1, OEO-1, AII-2, OEO-3, IAI-2, AII-4, AAI-2, AAO-2, EEI-2, EEO-2, EEI-3, EEO-3, EEI-4, EEO-4, EEI-1, EEO-1, IIO-1, IIO-2, IIO-3, IIO-4, IEO-1, IEO-2, IEO-3, IEO-4, OII-1, OOI-1, IOI-3, IOO-3, OII-3, IOI-4, IOO-4, OOI-4, OOO-4, EOI-1, EOI-2, OEI-4, OEI-2, OEO-2, OEO-4, AEI-1, AEO-1, AAO-3, AEI-3, AEO-3, EAI-3, EAI-4, OOI-3, AOO-1, IAO-1, OAI-1, OEI-1, IAO-2, EOI-3, OEI-3, AOI-4, EOI-4, AOI-2, OAI-2, OAO-2, IEI-1, EII-1, EII-2, IEI-2, EII-3, IEI-3, EII-4, IEI-4 |
| Certainly; true; ratio=1.0 | 25 | AAA-1, AAI-1, AII-1, EAE-1, EAO-1, EIO-1, AEE-2, AEO-2, AOO-2, EAE-2, EAO-2, EIO-2, AAI-3, AII-3, EAO-3, EIO-3, IAI-3, OAO-3, AAI-4, AAO-4, AEE-4, AEO-4, EAO-4, EIO-4, IAI-4 |

Matching Multilingual Tags Based on Community of Lingual Practice from Multiple Folksonomy: A Preliminary Result

Jason J. Jung

Knowledge Engineering Laboratory
Department of Computer Engineering
Yeungnam University
Gyeongsan, Korea 712-749
j2jung@{ynu.ac.kr,intelligent.pe.kr}

Abstract. By taking into account various co-occurrence patterns from a folksonomy, semantic correspondences between tags have been discovered and applied to a number of applications (e.g., recommendation). In this paper, we propose a novel collective intelligence application for expanding and transforming queries for searching for multilingual resources. Thereby, multilingual tags (e.g., between ‘Seoul’ in English and ‘Coree’ in French) within a folksonomy have been analyzed whether they have a significant relationship or not. We have tested the proposed multilingual tag matching method by collecting real-world tagging information from several well-known social tagging websites (e.g., Del.icio.us), and applied to translating queries to other languages without any external dictionary.

Keywords: Collective intelligence, Tag matching, Multilingual tagging, Semantic grounding, Social tagging, Folksonomy.

1 Introduction

Tags are collected from end users in various information spaces, e.g., blogs and wikis. They are called folksonomies (in other words, collaborative or social taggings), and regarded as an important evidence to implement a collective intelligence (CI) [1]. Moreover, a number of interesting applications and services have been designed and developed in a form of open APIs. They can be combined with each other for better services, so-called mashup [2,3].

However, more crucial challenge on these folksonomies is to discover hidden patterns between knowledge provided from users underlying the online information space. It means that individual intelligence of each user has been reflexed into the set of tags, and it is necessary to integrate the individual intelligence with others for solving more complex problems [4,5].

Particularly, we focus on taggings written in different languages in a number of information spaces. In this work, we focus on multilingual folksonomy to show what kind of (and how) the CI can be emerged *i)* among users who speak different languages and *ii)* among multilingual users. For example, in Del.icio.us, a

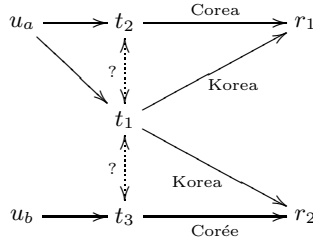


Fig. 1. Multilingual folksonomy system with three different languages (i.e., English, Spanish, and French). Two users u_a and u_b have tagged two resources r_1 and r_2 with three tags t_1 , t_2 , and t_3 .

bookmark may be tagged with multilingual tags at the same time, as shown in Fig. 1. Given certain bookmarks (e.g., <http://www.korea.net/>), in the first case, both a tag “Korea” by Jason and a tag “Corée” by Jérôme can be attached together. As second case, if Jason is a bilingual person (e.g., English and Spanish), he might use two tags “Korea” and “Corea”, simultaneously.

There have been many studies to discover meaningful information (e.g., tag relatedness [6]) from a given folksonomy. However, most of them are limited on an identical language. Thus, we can note that in this work there are two main issues to deal with as follows;

- how to reveal relationships between multilingual tags (i.e., between t_1 and t_2 and between t_1 and t_3) and between users (i.e., u_a and u_b), and
- how to exploit the discovered information for better services (i.e., searching for r_1 to u_b).

Thereby, we want to identify individual intelligence (i.e., ability to speak languages and to translate from one to another) by social affinity between users participating in the folksonomies. Finally, community of lingual practice can be organized for providing users with better services (e.g., query transformation [7] and mashup [2]). More importantly, social reputation has been exploited to identify individual intelligence from multiple folksonomies. We may regard the social reputation as an indirect measurement of each user’s expertise on multilingual skills and practice.

Furthermore, as another important contribution of this study, we have to mention that background knowledge with external sources (e.g., dictionaries and thesauri) is not needed to refer to. Also, some words newly generated from online can be efficiently matched.

The outline of this paper is as follows. In Sect. 2, we explain several definitions with notations, and how to conduct semantic grounding of the tags by using such notations. Sect. 3 address social consensus-based identification of the lingual practice of online users. Then, in Sect. 4, we want to show how to use the matching results between multilingual tags on tag-based information retrieval systems. Sect. 5 demonstrates experimental results obtained from the tag-based information retrieval systems. In Sect. 6 and Sect. 7, we will discuss and compare

several important issues from the proposed method with some existing ones, and finally, draw a conclusion of this work.

2 Semantic Grounding of Tags

In this section, we want to discuss formalization as well as to semantic grounding of the tagging information from users.

Definition 1 (Tag). *A certain user u can annotate a set of tags t_u^r for describing his context about a certain resource r . A tag set by u can be represented as*

$$T_u = \{(t \times r \times l) | l \in \mathbb{L}_u\} \quad (1)$$

where t indicates a keyword written in a language $l \in \mathbb{L}_u$.

Definition 2 (Folkosonomy). *A folkosonomy \mathbb{F} can be simply written by*

$$\mathbb{F} = \bigcup_{u_a \in U_{\mathbb{F}}} T_{u_a} \quad (2)$$

where $U_{\mathbb{F}}$ is a set of users contributing their intelligence to the folkosonomy. It is a simple integration of their sets of tags. In addition, we can easily compute the number unique resources (denoted as $|\mathbb{R}_{\mathbb{F}}|$).

For instance, in Fig. 1, the tag sets of two users u_a and u_b can be given by

$$T_{u_a} = \{\langle \text{Korea}, r_1, EN \rangle, \langle \text{Korea}, r_2, EN \rangle, \langle \text{Corea}, r_1, SP \rangle\} \quad (3)$$

$$T_{u_b} = \{\langle \text{Corée}, r_2, FR \rangle\} \quad (4)$$

where the tags are written in three different languages.

Here, we want to discover the co-occurrence patterns inside the folkosonomy \mathbb{F} . Two kinds of functions τ_U and τ_R are formulated to measure *i*) lingual practices of an individual user (Δ_U), and *ii*) the co-occurrence patterns between users (Δ_R), respectively.

Definition 3 (Δ_U -Co-occurrence pattern). *Given a certain user u_a , languages used for multilingual tags can be found out. Thus, for all his tags, his lingual practice can be represented by a function*

$$\tau_U(l, l') = \frac{|\{t | (t \times r \times l), (t' \times r \times l') \in T_{u_a}, l \neq l'\}|}{\max(|\{t | t \times l\}|, |\{t' | t' \times l'\}|)} \quad (5)$$

where $l, l' \in \mathbb{L}_{u_a}$ and $t, t' \in T_{u_a}$.

This pattern means how frequently he has been using multilingual tags with a certain resource at the same time.

Definition 4 (Δ_R -Co-occurrence pattern). *Given two sets of resources \mathbb{R} , co-occurrence patterns between multilingual tags can be found by a function*

$$\tau_R(t, t') = \frac{|\{\langle t, t' \rangle | \langle t \times r \times l \rangle \in T_{u_a}, \langle t' \times r \times l' \rangle \in T_{u_b}\}|}{\max(|\mathbb{R}_{\mathbb{F}_a}|, |\mathbb{R}_{\mathbb{F}_b}|)} \quad (6)$$

where $|\mathbb{R}_{\mathbb{F}_a}|$ and $|\mathbb{R}_{\mathbb{F}_b}|$ are the numbers of the resources from folksonomies the corresponding users are participating in.

Regarding the folksonomies for Δ_U - and Δ_R -Co-occurrence patterns, it is not necessary to be same information sources of such folksonomies. It means that in this work, we want to merge partial intelligence extracted from each folksonomy into one.

3 Identifying Individual Intelligence

Here, we want to address that online collaborative systems (e.g., folksonomies) need to be more robust. The system can not guarantee that all of anonymous users provide reliable information and contribute rational efforts on online systems. Thus, we simply assume that the lingual practice of each user will be more expertise when more people have the same opinions with his.

Definition 5 (Social consensus). *Given a user u_a , a social consensus about his lingual practice between two languages l and l' is represented by*

$$\phi_{u_a}^{l, l'} = \tau_R(t, t') \times \left(\exp \frac{|\mathbb{U}^+|}{|\mathbb{U}|} - 1 \right) \quad (7)$$

where $|\mathbb{U}^+|$ means the number of online users who has same opinion with u_a out of the whole participants \mathbb{U} .

Once we have this social consensus of all users, we can build community of lingual practice \mathbb{C} . With respect to two languages in Equ. 7, a group of users whose $\phi^{l, l'}$ is over a pre-defined threshold can be selected as a community member $u_a \in \mathbb{C}^{l, l'} \subseteq \mathbb{C}$. In fact, the number of all possible communities is depending on the number of languages used by people (i.e., $|\mathbb{C}| = |\mathbb{L}| C_2$). More importantly, the value of $\phi_{u_a}^{l, l'} \times \tau_U(l, l')$ can be regarded as a centrality of user u_a within the corresponding community.

As a result, we can discover matching between multilingual tags, which are called correspondences.

Definition 6 (Correspondence). *A set of correspondences discovered matching process between two multilingual tags is given by*

$$\bigcup_{\mathbb{C} \supseteq} \{\langle t, t' \rangle | \langle t \times r \times l \rangle, \langle t' \times r \times l' \rangle \in T_{u_a}, u_a = \max_{u \in \mathbb{C}^{l, l'}} u\} \quad (8)$$

where $\mathbb{C} \supseteq$ indicate any possible communities.

4 Tag-Based Information Retrieval

The goal of the proposed multilingual tag analysis is basically to match two multilingual tags. Here, we want to demonstrate one possible application, tag-based information retrieval, within multiple folksonomies, differently from Tagster [8]. Similar to query translation [9], a tag-based query can be transformed (translated) into another language, by referring to the correspondences.

Definition 7 (Tag-based query). *A tag-based query q can be assumed to be travelled from u_{src} to u_{dest} . The query grammar is simply given by*

$$q ::= t_{src} | \neg q | q \wedge q | q \vee q \quad (9)$$

where $t_{src} \in T_{src}$.

In this study, we are interested in queries consisting of a set of tags written by the language of source user, so that the queries can be transformed by *tag replacement* strategy based on correspondences discovered by multilingual tag matching. The basic idea of this strategy is as follows.

1. Making member list by sorting out the community member with respect to the centrality value
2. Finding the correspondences from the member list until replacing all of tags in the query

Unless we can find all the matches, the query transformation process returns failure by nature. In addition, we want to note that the community organized by the folksonomy can be overlapped. It means that the users can be included in more than one community.

5 Experimentation

To evaluate the proposed scheme, we have collected a large amount of tag information from the following information sources;

- Social bookmarking systems, Del.icio.us¹
- Photo sharing systems, Flickr²

More importantly, as a language identifier, we have exploited a Google AJAX Language API³. This API can identify more than 20 languages.

5.1 Evaluation of Multilingual Tag Matching

Due to the lack of tags, we have chosen only 12 languages to test the proposed scheme. We did not implement any preprocessing procedure for stop word removal and stemming. By comparing with normal dictionaries, a precision factor ($P(l, l')$) has been measured, but a recall factor has not.

¹ <http://del.icio.us/>

² <http://www.flickr.com/>

³ <http://code.google.com/apis/ajaxlanguage/documentation/reference.html>

Table 1. Precision of multilingual tag matching (%)

| | l_1 | l_2 | l_3 | l_4 | l_5 | l_6 | l_7 | l_8 | l_9 | l_{10} | l_{11} | l_{12} |
|----------|-----------|-----------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|
| l_1 | - | - | - | - | - | - | - | - | - | - | - | - |
| l_2 | 67 | - | - | - | - | - | - | - | - | - | - | - |
| l_3 | 56 | 68 | - | - | - | - | - | - | - | - | - | - |
| l_4 | 35 | 84 | 73 | - | - | - | - | - | - | - | - | - |
| l_5 | 87 | 57 | 54 | 45 | - | - | - | - | - | - | - | - |
| l_6 | 45 | 83 | 66 | 56 | 74 | - | - | - | - | - | - | - |
| l_7 | 48 | 63 | 73 | 52 | 68 | 62 | - | - | - | - | - | - |
| l_8 | 82 | 67 | 54 | 62 | 62 | 64 | 63 | - | - | - | - | - |
| l_9 | 65 | 53 | 34 | 53 | 73 | 72 | 45 | 43 | - | - | - | - |
| l_{10} | 47 | 65 | 51 | 43 | 42 | 34 | 78 | 51 | 58 | - | - | - |
| l_{11} | 75 | 73 | 61 | 66 | 63 | 72 | 62 | 63 | 52 | 62 | - | - |
| l_{12} | 72 | 23 | 72 | 75 | 32 | 43 | 43 | 43 | 62 | 73 | 82 | - |

As shown in Table 1. The average precision of tag matching between all possible combination is 59.78%. While the maximum precision is about 87.2%, which is the tag matching between French and Japanese, the minimum precision is about 23.3% between Czech and Polish.

5.2 Evaluation of Multilingual Resource Retrieval

We have selected a number of users to organize a simplified community of lingual practice. Hence, by using community identification equation, we have organized three communities with highly cohesive community, as follows.

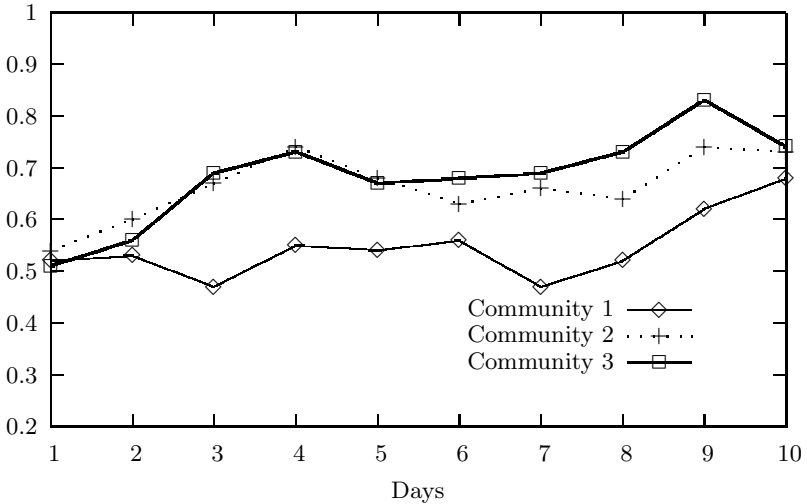


Fig. 2. Measuring user satisfaction with RSS feeds via the proposed tag-based information retrieval system

- Community₁: U₁, U₈, U₁₀, U₁₂, U₁₄ (i.e., $Q^\diamond(\text{Community}_1) = 0.792$)
- Community₂: U₂, U₃, U₆, U₇, U₁₅, U₁₆, U₁₇ (i.e., $Q^\diamond(\text{Community}_2) = 0.817$)
- Community₃: U₄, U₅, U₉, U₁₁, U₁₃, U₁₈ (i.e., $Q^\diamond(\text{Community}_3) = 0.792$)

We found out that the proposed community identification method is working very well, because the modularity value is relatively high. In particular, although Community₂ have more members, the cohesiveness is higher than other communities.

As second experimentation, we have conducted human evaluation for RSS-based information recommendation. During 10 days, we kept tracking of bloggers' rating patterns, as shown Fig. 2. The precision (i.e., user satisfaction) of context-based RSS feeds in three communities has been increased over time in common. This is caused by the dynamic community identification process. The bloggers were able to be automatically involved into more relevant communities.

6 Discussion

We want to mention several important issues that we have realized from this work. Firstly, Comparison with many studies on folksonomy analysis is needed. In terms of using co-occurrence, which can be regarded as social consensus by simple statistics, the proposed approach is similar to them. However, we have tackled analyzing multilingual tagging correspondences. Also, we have tried to identify single intelligence of each user from his lingual practice. Therefore, the precision measures in both of matching multilingual tags and providing better services have shown significantly high level.

Even though there have been a number of research investigations on information retrieval and natural language processing communities, very few studies on resources with multilingual annotations (e.g., tags) have been done (e.g., e-learning resources [10]). As additional implication, we found out the users who speak quite different language families (e.g., French and Japanese) tend to contribute more multilingual tags, and be better performance.

Furthermore, some works on semantic web and knowledge engineering communities have been proposing some standardization in a form of ontologies and XML-based metadata [11]. In our study, we have designed automated data portability between multiple folksonomies without any metadata.

7 Concluding Remarks and Future Work

In this paper, we have proposed two main contributions; *i*) representation of tagging contexts (i.e., why the users are using the tags), and *ii*) co-occurrence measurement between multilingual tags. As a conclusion, this paper have shown higher performance of multilingual query transformation and information retrieval, without exploit description about tag context. It was about 60% precision.

However, we have realized that the Google language API we have exploited have quite high rate of error result for language identification. We can expect that we will have a far higher performance when the API provide more robust identification.

Also, as a future work, we are planning

- to employ natural language processing tools to preprocess the user tags, (because there are too many typo errors and mistakes from end users)
- to combine more folksonomies which are available on the web, and
- to consider user identification in different folksonomies by using a certain identity indicators (e.g., email address, Open ID, and so on).

Acknowledgement

This work was supported by the Science and Technology Amicable Research (STAR) by Korean NRT grant funded by the Korean government (MEST). (No. 2009-50252).

References

1. Jung, J.J.: Knowledge distribution via shared context between blog-based knowledge management systems: a case study of collaborative tagging. *Expert Systems with Applications* 36(7), 10627–10633 (2009)
2. Pautasso, C.: Restful web service composition with bpel for rest. *Data & Knowledge Engineering* 68(9), 851–866 (2009)
3. Granell, C., Díaz, L., Gould, M.: Service-oriented applications for environmental models: Reusable geospatial services. *Environmental Modelling & Software* 25(2), 182–198 (2010)
4. Lévy, P.: *Collective Intelligence: Mankind’s Emerging World in Cyberspace*. Basic Books, New York (1994)
5. Pentland, A.: On the collective nature of human intelligence. *Adaptive Behavior* 15(2), 189–198 (2007)
6. Cattuto, C., Benz, D., Hotho, A., Stumme, G.: Semantic grounding of tag relatedness in social bookmarking systems. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) *ISWC 2008*. LNCS, vol. 5318, pp. 615–631. Springer, Heidelberg (2008)
7. Jung, J.J.: Query transformation based on semantic centrality in semantic social network. *Journal of Universal Computer Science* 14(7), 1031–1047 (2008)
8. Görlitz, O., Sizov, S., Staab, S.: Tagster - tagging-based distributed content sharing. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) *ESWC 2008*. LNCS, vol. 5021, pp. 807–811. Springer, Heidelberg (2008)
9. Sheridan, P., Ballerini, J.P.: Experiments in multilingual information retrieval using the SPIDER system. In: Frei, H.P., Harman, D., Schäuble, P., Wilkinson, R. (eds.) *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1996)*, Zurich, Switzerland, pp. 58–65. ACM, New York (1996)
10. Vuorikari, R., Ochoa, X., Duval, E.: Analysis of user behavior on multilingual tagging of learning resources. In: *Proceedings of the 1st Workshop on Social Information Retrieval for Technology-Enhanced Learning*, pp. 6–17 (2007)
11. Kim, H.L., Scerri, S., Breslin, J.G., Decker, S., Kim, H.G.: The state of the art in tag ontologies: a semantic model for tagging and folksonomies. In: *Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications (DCMI 2008)*, pp. 128–137. Dublin Core Metadata Initiative (2008)

Multiclass Mineral Recognition Using Similarity Features and Ensembles of Pair-Wise Classifiers

Rimantas Kybartas¹, Nurdan Akhan Baykan², Nihat Yilmaz³, and Sarunas Raudys¹

¹ Department of Computer Science, Vilnius University, Vilnius, Lithuania

² Department of Computer Engineering, University of Selcuk, Konya, Turkey

³ Department of Electric-Electronics Engineering, University of Selcuk, Konya, Turkey

rimantas.kybartas@mif.vu.lt, {nurdan,nyilmaz}@selcuk.edu.tr,
raudys@ktl.mii.lt

Abstract. Mineral determination is a basis of the petrography. Automatic mineral classification based on digital image analysis is getting very popular. To improve classification accuracy we consider similarity features, complex one stage classifiers and two-stage classifiers based on simple pair-wise classification algorithms. Results show that employment of two-stage classifiers with proper parameters or K class single layer perceptron are good choices for mineral classification. Similarity features with properly selected parameters allow obtaining non-linear decision boundaries and lead to sizeable decrease in classification error rate.

Keywords: Mineral classification, single layer perceptron, support vectors, two stage classifiers, similarity features.

1 Introduction

Color is a fundamental physical property of image processing. It is widely used in physical analysis [1], [2], [3]. In optical mineralogy, color is used for the recognition of minerals in order to identify the rock names. Color is useful for the recognition of minerals under microscopes with polarized light. Hence, microscopes with polarized light capabilities are used for optical mineralogy [4], [5], [6], [7].

Microscopes are commonly used for manual mineral identification in thin sections. But there are some problems about color which depend on a variety of factors including illumination, mineral type, thickness of thin section etc. Thus, automated mineral identification systems [5], [6] are based on scanned or plane and polarized images and use the natural color of the mineral.

Today, many vision systems appear for the quality control of products in all areas [8]. They have been applied for boundary detection, segmentation, feature extraction and identification of objects. Because of these varieties of applications, image vision is getting popular and also is used in different fields [9], [10], [11]. In this study, thin section images were analyzed by using image processing in order to identify minerals.

After getting color parameters of minerals, they have to be passed to some classification system in order to know which class of minerals they represent. A lot of research

has been done on mineral exploration, e.g. [5], [6] [7]. Most of researchers paid principal attention to obtaining of data, its preprocessing and proper feature selection. But they missed a thorough analysis of experiment performance. That could lead to inaccurate results. The most frequently used method for mineral classification is artificial neural network (ANN) with one hidden layer [5], [7]. But its shortcoming is that selection of proper architecture (i.e. selecting best amount of neurons in each layer) is time consuming.

The objective of this study is to find simple, reliable method suitable for mineral data classification. In this study we consider two types of classification methods – standard multiple class classifiers and two stage classifiers based on simple pair-wise classifiers. The last ones were selected due to their lightweight, fixed architecture and proven ability to perform not worse or even better than standard ANNs solutions.

The mineral data distributions we analyze in this study are rather complicated – classes have highly diverse sizes and covariance matrixes. Besides, classes are overlapping and located near each other. Thus we employed similarity features in order to separate data and to get higher classification performance.

The paper is organized in the following way. In sections 2 and 3 we present the classification methods we used. In section 4 we present obtaining and manipulation of mineral data, while obtained results are listed in section 5. Conclusions and remarks are in section 6.

2 One Stage Classifiers

One stage classification methods are methods where only one algorithm is employed and used for decision.

Multi-class single layer perceptron (K-SLP). Multi-class single layer perceptron [12], [13] consists of one layer containing K single layer perceptrons $f(\mathbf{w}^T \mathbf{x} + b)$ where \mathbf{x} is data vector, \mathbf{w} is a p -dimensional weight vector, b is a bias weight and f is the activation function. In our study we used nonlinear sigmoid function

$$f(x) = 1/(1 + e^{-x}) \quad (1)$$

as the activation function.

Radial basis function (RBF) neural networks. The RBF-based classifier consists of three layers: input layer, a hidden radial basis function layer and a linear output layer [12], [13], [14], [15]. Radial basis layer is composed of g radial basis neurons that calculate $y_i = \text{rad}(\|C_{i1} - x\|/H_i)$, $i = 1, \dots, g$. We used the model of multivariate Gaussian distribution as a transfer function for radial basis neuron *rad*. C_{i1} is the i -th “center” of the radial basis neuron, and H_i is the smoothing parameter. Output layer is linear: $o_j = \mathbf{w}_{j2}^T \mathbf{y} + b_{j2}$, where $\mathbf{y} = (y_1, \dots, y_g)^T$, \mathbf{w}_{j2} is the weight vector and b_{j2} is the bias term. The newly classified vector \mathbf{x} is classified according to the maximum of outputs. We used the Matlab neural network (NN) toolbox to form the RBF networks. Artificial pseudo-validation sets were used to select parameters g and H_i (see the end of section 3.1).

Kernel discriminant analysis (KDA). In KDA approach, conditional probability density functions of input vectors are used. Then the kernel-based local estimates are applied to their results. In classification phase, independent decisions are performed at each point of the feature space [13], [15]. We used the Gaussian kernel and classified according to the maximum of products

$$\frac{q_i}{N_i} \sum_{j=1}^{N_i} \exp(-h^{-1}(\mathbf{x} - \mathbf{x}_{ij})^T (\mathbf{x} - \mathbf{x}_{ij})) \quad (2)$$

where q_i is prior probability of the class $i = 1, 2, \dots, K$; $j = 1, 2, \dots, N_i$, K – number of classes, N_i – number of training vectors in each class, and h is a smoothing parameter (we used value 1.0 due to data normalization).

3 Two Stage Classifiers

In two stage classification methods, the first stage uses some classifiers for primary classification and then, in the second stage, the results of this primary classification are fused in a particular manner.

3.1 Pair-Wise Classifiers

In this study we used pair-wise classifiers classifying only two classes as classifiers in the first stage. Thus in the case of classifying K data classes, $K(K-1)/2$ pair-wise classifiers are obtained. In our study we used single layer perceptrons (SLPs) [12], [13] and support vector classifiers (SVCs) [16] as pair-wise classifiers.

Single layer perceptron. Single layer perceptron (SLP) is an artificial model of biological neuron which may be represented as $f(w^T x + b)$, where w and b are weight vector and bias which are obtained during process of perceptron training. Vector x is a p -dimensional data vector and f is the output activation function (1).

We used gradient descent algorithm [12] to train SLP. The attractive feature of SLP is that during its training it may evolve through seven different statistical classifiers [17] which may be optimal classifiers for data with particular properties. Usually these properties of the data are not known. Thus it is very important to stop SLP training at the proper time. One of the best ways to do that is to use pseudo-validation data [13] (see the last subsection of this section).

Support vector classifier. Support vector classifier (SVC) [16] solves the following primal problem:

$$\min_{w, b, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (3)$$

subject to $y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$, $\xi_i \geq 0$, $i=1, \dots, l$.

Where $y_i \in \{1, -1\}$ is the label of one of l training vectors x_i , $C > 0$ is the upper bound and $\phi(x_i)$ is a kernel function. Since SLP in the classification task is a linear function, i.e. result depends on the value of the argument of function (1), in order to make fair comparison we chose kernel function of SVC to be $\phi(x_i) = x_i$. Thus we used weighted linear SVC realized in *LIBSVM* library [18]. We used pseudo-validation data (see next subsection) to select proper upper bound C from value set $[2^{-7}, 2^{-6}, \dots, 2^{10}]$ [19]. We also employed different weighting of parameter C for each class of the pair. Weights were selected inversely proportional to the number of training vectors in that class [20].

Pseudo-validation data. The number of iterations for SLP training, parameter C for SVC, parameters of RBF neural network and parameters of similarity features (see section 4) were selected according to classification error on pseudo-validation data. It was formed from training data by means of k nearest neighbors colored noise injection [21]. To generate such noise, for each single training vector x_i , we find its k nearest neighbors of the same pattern class and add an artificially generated noise only in a subspace formed by the vector x_i and k neighboring training vectors $x_{i1}, x_{i2}, \dots, x_{ik}$. Random Gaussian $N(0, \sigma^2)$ variables are added ni_{nm} times along the k lines connecting vector x_i and $x_{i1}, x_{i2}, \dots, x_{ik}$. The noise injection parameters were chosen empirically: $k=2$, $\sigma=1.0$ and $ni_{nm}=2$.

3.2 Fusion Methods

After obtaining decisions of pair-wise classifiers, one has to properly fuse their results. There are plenty of methods dedicated to this (e.g. [22 – 26]). In this study we chose popular, simple methods.

Voting. This rule performs allocation of $K(K-1)/2$ -dimensional vector formed by the first stage pair-wise classifiers according to the majority of class labels in this vector. This method is also known as “Max Wins” method.

The directed acyclic graph method (DAG). This algorithm (DAGSVM [27]) organizes pair-wise SVMs in rooted binary direct acyclic graph to make the final decision. When a vector is submitted for classification, it is first evaluated by the root classifier (root DAG node). Subsequently, decision making is passed to the left or right node depending on the current node decision until one of K nodes with no children is reached. This node labels a new vector. In our experiments, we also used pair-wise SLPs instead of SVMs, used in original paper.

Hastie-Tibshiranie (H-T) method. Let p_1, p_2, \dots, p_K be the probabilities of the vectors to be classified. An assumption is made that for each class pair i, j , $i \neq j$ there are n_{ij} data samples from which conditional probabilities $r_{ij} = \text{Prob}(i|j \text{ or } j)$ could be estimated. The proposed model in [28] is $\mu_{ij} = p_i / (p_i + p_j)$ in which they try to find such \hat{p}_i that $\hat{\mu}_{ij} = \hat{p}_i / (\hat{p}_i + \hat{p}_j)$ are close to r_{ij} . The closeness is determined by using criterion of average (weighted) Kulback-Leibler distance between r_{ij} and μ_{ij}

$$l(p) = \sum_{i < j} n_{ij} \left[r_{ij} \log \frac{r_{ij}}{\mu_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \mu_{ij}} \right] \quad (4)$$

and finding p to minimize this function. The score (gradient) equations are

$$\sum_{j \neq i} n_{ij} \mu_{ij} = \sum_{j \neq i} n_{ij} r_{ij} : i = 1, 2, \dots, K \quad (5)$$

subject to $\sum p_i = 1$. In order to compute \hat{p}_i authors of H-T method propose iterative procedure [28].

4 Data

In this study thin sections were observed using the James Swift microscope. Images were taken by a digital camera in a rotating experimental stage instead of a fixed stage. The experimental stage can be rotated from 0 to 180 degrees by 1 degree increments, while polarizer and analyzer remain crossed to each other in a vertical direction during the analysis. Illumination source was a 12V/100W halogen light. Thin section images were taken through a Videolab camera mounted on the microscope. Images were transmitted to a computer by Inca software.

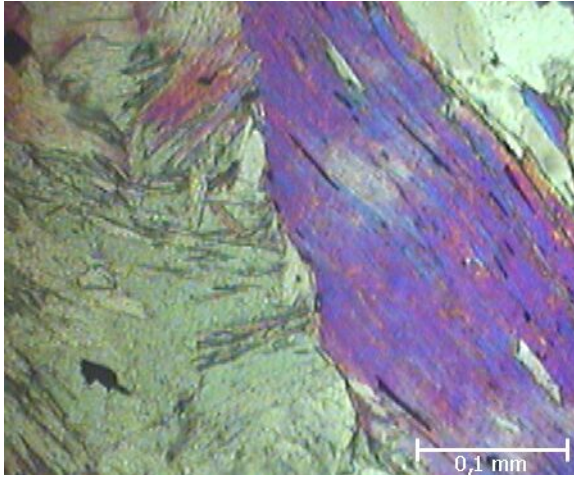


Fig. 1. Maximum intensity image of minerals

Images obtained both under plane-polarized and cross-polarized light contain the maximum intensity values (Fig.1). For maximum intensity the images at every 10 increment were first captured and then compared to the previous images. All images were stored in RGB format with the dimension of 450x370 pixels with the resolution of 150 dpi.

Twenty-two digital images were taken from nine thin sections. Thin sections were taken from the department of geological engineering in Selçuk University, Turkey. In our study, a total of 5 common minerals – quartz (110 samples), muscovite (110 samples), biotite (60 samples), chlorite (60 samples) and opaque (60 samples) – were used. For image quantization, first a median filter was applied to images for noise reduction and then the histogram was equalized. Thus we got 6 features of each mineral image pixel. The first three color parameters were extracted from cross-polarized light, and the other three from plane-polarized light.

Prior to training the classifiers, the data was normalized by standard deviations of each single feature. Then principal components and eigenvalues of pooled sample covariance matrix were used to transform the data prior to training the SLPs and SVCs. Moreover, prior to training the pair-wise classifiers, each time the two-class mean vectors were moved to the zero point.

Similarity features. In this study similarity features were employed. If x_i is an original data vector, then its similarity feature vector consists of N_{tr} (number of training vectors) components:

$$s_i^j = \exp(-\alpha * \sqrt{\sum_{k=1}^p (x_i^k - x_j^k)^2}), \quad (6)$$

where p is the dimensionality of data, j is the index of j th similarity feature, vector superscript k denotes k -th element of original data vector and α is the normalization coefficient. This way the new dimension equal to N_{tr} is obtained. In our study half of the 400 data vectors were used for training. Thus for each data vector x_i we obtained a 200-dimensional similarity feature vector s_i .

5 Results

In order to get a rather high reliability of results, we performed 250 2-fold cross validation generalization estimation procedures for all the data and all the methods. We permuted the data 250 times, divided it into two equal pieces for training and testing, and the same permutations of data were used in all kinds of experiments (different data and classification methods), i.e. all the experiments were performed with exactly the same data sets.

Firstly we employed all the classification algorithms with original data. The classification error rates were rather high. For the best pair-wise method (DAG + SVC as pair-wise classifier) it was 0.211 (see Table1). The multi-class RBF method, with error rate of 0.189, showed the best results among overall methods.

As in the studies done by other researchers, e.g. [5], [7], we also used ANN with one hidden layer for original mineral data. Neurons in hidden layer were selected (by employment of pseudo-validation data) from an empirically predefined set. The error rate obtained with such ANN was 0.25 – the worst out of the used methods. Besides, as it was already mentioned before, the selection of neurons in hidden layer was highly time consuming. Thus we refused to use this method for further study.

The effect of similarity feature employment may be seen in Fig. 2. The classes became “C” shaped and more separable.

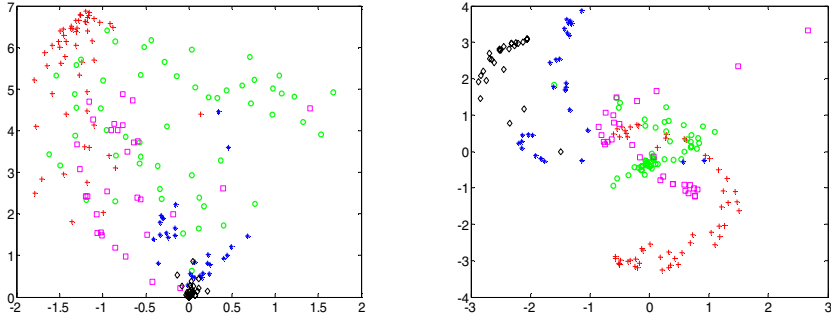


Fig. 2. Original data of five different minerals (left) and the same data after employment of similarity features (right). Dimensions were reduced according to eigenvalues of covariance matrixes (principal component analysis method). Same shapes and colors mean the same minerals.

Parameter α was selected for each classification method from empirically formed set of values [0.0001, 0.001, 0.01, 0.1, 0.2, 0.5, 0.8, 1.0, 1.2, 1.5, 1.8, 2] using pseudo-validation data (see section 3.1). The α was selected each time we used a new data permutation in training, i.e. 250x2 times.

Table 1. Results of mineral data classification. In the cells of two stage algorithm results, the left value shows the average error rate with SLP as a pair-wise classifier and the right one – with SVC as a pair-wise classifier. In the last two lines the most often selected similarity features' parameter α and its selection rate (from all experiments) are presented.

| Data parameters | One stage classifiers | | | Two stage classifiers | | |
|---------------------|-----------------------|-------|-------|-----------------------|-------------|-------------|
| | RBF | KDA | K-SLP | Voting | DAG | H-T |
| Original data | 0.189 | 0.226 | 0.252 | 0.212/0.211 | 0.215/0.211 | 0.226/0.218 |
| Similarity features | 0.177 | 0.212 | 0.174 | 0.227/0.174 | 0.238/0.174 | 0.173/0.183 |
| Best α | 10^{-4} | 0.1 | 0.5 | 0.1 | 0.1 | 0.5 |
| Best α rate | 0.63 | 0.83 | 0.47 | 0.86 | 0.86 | 0.54 |

The results presented on the second line of Table 1 show that despite the increase of dimensionality, multi-class one stage methods and pair-wise methods based on SVC showed better results than using original data.

The best method in our experiments was a two stage method based on SLP as pair-wise classifiers and using Hastie-Tibshirani fusion method (SLP+HT). For the estimation of inaccuracy the expression [13]

$$\sqrt{err(1 - err) / N_{ts}} \quad (7)$$

may be used, where err is the classification error rate and N_{ts} is the number of overall testing data vectors. Actually, in 2-fold cross validation all of the data is considered in one experiment (in one of the 2 cross validation sets). Thus despite having performed 250 experiments, generalization error was estimated on the same 400 data vectors we

had. So inaccuracy of estimation of generalization error of SLP+HT method is $\sqrt{0.173(1-0.173)/400} = 0.019$. Thus we may see that many other methods also performed well since their error rate fits within this accuracy interval (i.e. up to $0.173+0.019=0.192$). After employing similarity features, some multi-class methods performed better as well as pair-wise ones, but the latter ones are recommended for practical use due to their vast ability for further improvement.

The results with employment of pair-wise SLP in voting-based fusion methods (Voting and DAG) are not as good due to their sensitivity of sample size and dimensionality ratio. This behavior is due to their similarity to statistical methods [17] which work well if sample size is much greater than dimensionality because of problems arising with covariance matrix estimation (for more see e.g. [29]). From the results it may be also observed that probability estimation based H-T fusion method overcomes shortcomings of SLP classifier and exploits its results best.

In order to overcome high dimensionality problems for SLP as pair-wise classifier in voting based fusion methods we used simple linear dimension reduction using eigenvalues of covariance matrixes - principal component analysis. We reselected dimensionalities from vast set of different values from 2 to 195 (the total dimensionality of similarity features was 200). The experiments showed that with dimensionality reduced to proper size, better results may be obtained. E.g., when using dimensionality equal to 2, the results may become 1.5 times worse because of the loss of some information. While using dimensionality approximately between 40 and 50, SLP as pair-wise and Voting as fusion method (SLP+Voting) may perform with a generalization error rate of 0.176 – much better than without dimensionality modification (error rate of 0.227). If dimensionality is increased further, then classification error increases again due to additional redundant information. While reducing dimensionality the generalization errors of other methods also decreases, but within the above mentioned accuracy.

The results of this study also showed that parameter α used for obtaining similarity features highly depends both on classification method (see the best values in line 3 of Table 1) and data permutation (see the percentage of use of best α value in line 4 of Table 1).

Reliability of results. The number of experiments with different data permutations plays a great role in the reliability of results. If considering only one experiment, then out of 250 2-fold experiments using similarity features (without dimensionality reduction) we may select the best and the worst ones which are highly different from the means of all experiments listed in Table 1. E.g., in one experiment with SLP+Voting method we obtained an error rate of even 0.125, while with worst experiment for the best averaged methods (e.g. SLP+HT or K-SLP) we obtained an error rate of up to 0.25. So in order to obtain reliable results it is necessary to perform a rather large amount experiments.

Although the average results of our experiments do not seem perfect, they are satisfactory for the mineral classification, since an error rate of even 0.25-0.3 is acceptable for this task. Besides, the accuracy of classification may be increased by obtaining more pixels from the same mineral to be classified.

6 Conclusions and Remarks

In order to simplify complexity and obtain non-linear decision boundaries in the complex shaped input feature space we used similarity features. This data transformation makes data more separable. On the other hand it enlarges dimension and makes it hard for SLP pair-wise classifiers to learn due to their similarity to classical statistical classifiers. Thus for mineral classification we recommend using similarity features and two stage classification methods with pair-wise classifiers or fusion rules which are less sensitive to dimensionality increase or K-SLP multiple class classification method.

The main issue in classification of such complicated data is proper selection of parameters (both the classifier and the data). In order to get precise estimation of methods a lot of experimental results should be applied on as much data as possible.

Proposed two stage classification methods are much better in calculation speed than multilayer artificial neural networks, since they do not require special training procedure to obtain proper architecture (input values, number of hidden neurons).

The novelties of our study are: 1) using two-stage classification methods to classify multi-class mineral data, 2) using similarity features to make mineral data non-linearly more separable, and 3) using a pseudo-validation data set to select parameters of the decision making algorithms.

In this paper we used the straightforward way of using similarity features and dimension reduction by principal component analysis. The decision making strategy developed in present paper could be improved by using advanced similarity feature selection techniques. It also opens a free space for a straightforward introducing of pricing of incorrect classification, which is an important issue in industrial application.

References

1. Segnini, S., Dejmeck, P., Öste, R.: A low cost video technique for colour measurement of potato chips. *Lebensmittel-Wissenschaft und-Technologie* 32(4), 216–222 (1999)
2. Yam, K.L., Papadakis, S.E.: A simple digital imaging method for measuring and analyzing color of food surfaces. *Journal of Food Engineering* 61(1), 137–142 (2004)
3. Gökay, M.K., Gundogdu, I.B.: Color identification of some Turkish marbles. *Construction and Building Materials* 22(7), 1342–1349 (2008)
4. Fueten, F.: A computer-controlled rotating polarizer stage for the petrographic microscope. *Computers & Geosciences* 23(2), 203–208 (1997)
5. Marschallinger, R.: Automatic mineral classification in the macroscopic scale. *Computers & Geosciences* 23, 119–126 (1997)
6. Thompson, S., Fueten, F., Bockus, D.: Mineral identification using artificial neural networks and the rotating polarizer stage. *Computers & Geosciences* 27(9), 1081–1089 (2001)
7. Fueten, F., Mason, J.: An artificial neural net assisted approach to editing edges in petrographic images collected with rotating polarizer stage. *Computers & Geosciences* 33(9), 1176–1188 (2007)
8. Bombardier, V., Schmitt, E., Charpentier, P.: A fuzzy sensor for color matching vision system. *Measurement* 42(2), 189–201 (2009)

9. Komenda, J.: Automatic recognition of complex microstructures using the Image Classifier. *Materials Characterization* 46(2-3), 87–92 (2001)
10. Akesson, U., Stigh, J., Lindqvist, J.E., Göransson, M.: The influence of foliation on the fragility of granitic rocks, image analysis and quantitative microscopy. *Engineering Geology* 68(3-4), 275–288 (2003)
11. Forero, M.G., Stroubek, F., Cristobal, G.: Identification of tuberculosis bacteria based on shape and color. *Real-Time Imaging* 10(4), 251–262 (2004)
12. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford Univ. Press, Oxford (1995)
13. Raudys, S.: *Statistical and Neural Classifiers: An integrated approach to design*. Springer, NY (2001)
14. Haykin, S.: *Neural Networks: A comprehensive foundation*, 2nd edn. Prentice-Hall, Englewood Cliffs (1999)
15. Fukunaga, K.: *Introduction to Statistical Pattern Recognition*, 2nd edn. Academic Press, NY (1990)
16. Boser, B., Guyon, I., Vapnik, V.: A Training Algorithm for Optimal Margin Classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152. ACM Press, New York (1992)
17. Raudys, S.: Evolution and generalization of a single neurone. I. SLP as seven statistical classifiers, *Neural Networks* 11, 283–296 (1998)
18. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), Available at, <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
19. Hsu, C.-W., Chang, C.-C., Lin, C.-J.: *A Practical Guide to Support Vector Classification* (2009), <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
20. Wu, K.-P., Wang, S.-D.: A weight initialization strategy for weighted support vector machines. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) *ICAPR 2005*. LNCS, vol. 3686, pp. 288–296. Springer, Heidelberg (2005)
21. Skurichina, M., Raudys, S., Duin, R.P.W.: K-NN directed noise injection in multilayer perceptron training. *IEEE Trans. on Neural Networks* 11(2), 504–511 (2000)
22. Park, S.-H., Fürnkranz, J.: Efficient Pairwise 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. 658–665. Springer, Heidelberg (2007)
23. Fürnkranz, J.: Round Robin Classification. *Journal of Machine Learning Research* 2, 721–747 (2002)
24. Krzysko, M., Wolynski, W.: New variants of pairwise classification. *European Journal of Operational Research (EOR)* 199(2), 512–519 (2009)
25. Sulzmann, J.-N., Fürnkranz, J., Hüllermeier, E.: On Pairwise Naive Bayes Classifiers. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) *ECML 2007*. LNCS (LNAI), vol. 4701, pp. 371–381. Springer, Heidelberg (2007)
26. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. *J. Artif. Int. Res.* 2, 263–286 (1995)
27. Platt, J.C., Cristianini, N., Shawe-Taylor, J.: Large margin DAG's for multi-class classification. In: *Advances in Neural Information Processing Systems*, vol. 12, pp. 547–553. MIT Press, Cambridge (2000)
28. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. *The Annals of Statistics* 26(1), 451–471 (1998)
29. Friedman, J.H.: Regularized discriminant analysis. *Journal of the American Statistical Association* 84(405), 165–175 (1989)

Ensembles of Probability Estimation Trees for Customer Churn Prediction

Koen W. De Bock and Dirk Van den Poel

Department of Marketing
Faculty of Economics and Business Administration
Ghent University
Tweekerkenstraat 2, B-9000 Ghent, Belgium
{Koen.DeBock,Dirk.VandenPoel}@UGent.be
<http://www.crm.ugent.be>

Abstract. Customer churn prediction is one of the most important elements of a company's Customer Relationship Management (CRM) strategy. In this study, two strategies are investigated to increase the lift performance of ensemble classification models, i.e. (i) using probability estimation trees (PETs) instead of standard decision trees as base classifiers, and (ii) implementing alternative fusion rules based on lift weights for the combination of ensemble member's outputs. Experiments are conducted for four popular ensemble strategies on five real-life churn data sets. In general, the results demonstrate how lift performance can be substantially improved by using alternative base classifiers and fusion rules. However, the effect varies for the different ensemble strategies. In particular, the results indicate an increase of lift performance of (i) Bagging by implementing C4.4 base classifiers, (ii) the Random Subspace Method (RSM) by using lift-weighted fusion rules, and (iii) AdaBoost by implementing both.

Keywords: CRM, database marketing, churn prediction, PETs, probability estimation trees, ensemble classification, lift.

1 Introduction

In today's business environment, an effective Customer Relationship Management strategy is of the foremost importance [1]. One of the most important aspects of CRM is customer retention, i.e. the prevention of customers from ceasing to buy products or services and leaving the company. One often-used strategy in this context is to identify the potential churners in an early stage, and to treat these customers accordingly by offering adapted incentives in order to re-establish their relationship with the company. This practice is generally pursued in churn prediction.

In churn prediction, information from customers that is available in the company database is used to determine their proneness to attrite. Relevant predictive features typically include historical transactions from the customer with the company, demographic information of the customer and so on. Data mining techniques, and more specifically, classification algorithms, are then deployed to generalize the relationship that exists between a customer's characteristics, and his or her probability to churn [2]. Once built, these models can be used to predict the future behavior of customers and to deliver targeting information for churn-preventing marketing campaigns.

In a churn-prediction model, predictive performance is extremely important [3]. In this study, the use of ensemble learning for churn prediction is considered. Applications of ensemble classifiers to churn prediction include Random Forests [4], AdaBoost [5], AdaCost [6], Bagging [7], Stochastic Gradient Boosting [8] and ensembles of Artificial Neural Networks [3]. These studies all demonstrate the beneficial impact of using ensemble classifiers over single classifiers for classification performance in the context of churn prediction.

An often-used performance criterion in churn prediction is lift (e.g. [7,8]). Lift measures how many times the classification model improves the identification of potential churners in the selection, over random guessing. A popular criterion in research on churn is top-decile lift, where the top 10 percent of customers with the highest probabilities to churn are considered. In this study, two strategies to improve lift performance of four well-known ensemble classifiers are presented. A first strategy involves using C4.4 probability estimation trees (PETs) [9] as base classifier in the ensemble classifiers instead of regular C4.5 decision trees [10]. Probability estimation trees are designed to generate better posterior probabilities than regular decision trees. They have been shown to provide better ranking capabilities than regular decision trees, and can hence be expected to improve lift performance when used in ensembles. A second strategy involves altering the fusion rules that are used to combine the predictions of individual ensemble member classifiers into aggregate predictions. A comparison is made between average aggregation, which is considered to be the most basic fusion rule when rankable predictions are desired, to weighted approaches based on lift performance measures of the individual classifiers in the ensemble. In an experimental validation, the effect of both strategies will be investigated for Bagging [11], the Random Subspace Method (RSM; [12]), the combination of Bagging and RSM (SubBag; [13]) and AdaBoost [14]. Experiments are conducted for five real-life churn data sets from various European companies, belonging to different sectors. Also, in addition to top-decile lift, a range of alternative selection percentages is considered for the calculation of lift measures.

The paper is organized as follows. In Section 2, an overview is presented of probability estimation trees, ensemble classification and the ensemble classifiers considered in this study, and lift. Section 3 presents an overview of the used churn data sets, the conditions and the results of an experimental comparison. In a final section, a conclusion is formulated, and limitations to the study and directions for future research are provided.

2 Methodology

2.1 Probability Estimation Trees

Tree induction algorithms, such as C4.5, CART and CHAID are primarily designed to generate 'crisp' classifications: they map an instance, based on its values on a set of features, to precisely one class. However, many applications, such as churn prediction benefit from the availability of an estimation of confidence in a class prediction, such as a class membership probability. An alternative to standard classification trees in this aspect are probability estimation trees (PETs) that estimate class membership probabilities. Many PETs have been introduced in literature (e.g. [9][15]). In its most basic form, maximum likelihood probability estimates are generated as follows. Denote a decision tree that represents a C -class classification problem. Assume that at the end leaf for class $c_i \in \{c_1, c_2, \dots, c_C\}$, there are N instances belonging to $C_l \leq C$ classes, and that k instances belong to class c_i . The maximum likelihood posterior class membership probability for class c_i is then equal to $\frac{k}{N}$ [16]. It has been shown that PETs based on this rule often perform poorly at estimating class membership probabilities [9][17]. In [9], Provost and Domingos identify a number of reasons for this. They argue that maximum likelihood probabilities are potentially highly inaccurate, especially if the number of training instances at an end leaf is small. Further, they argue that pruning, aimed at constructing small but accurate trees, results in lower quality of estimated class probabilities. In order to generate better PETs, they suggest C4.4, an adapted version of the C4.5 tree-building algorithm. In C4.4, maximum-likelihood estimates are smoothed by using the Laplace correction, which adjusts probability estimates in order to make them less extreme. The Laplace estimate used for C4.4 is given by $\frac{k+1}{N+C_l}$. Further, in C4.4, no pruning is applied, and 'collapsing', a secondary pruning strategy inherent to C4.5, is no longer performed.

In [9], Provost and Domingos applied Bagging to C4.4 PETs and demonstrated in their experiments that an ensemble of PETs substantially improved AUC performance for a majority of the examined data sets. However, they did not consider alternative ensemble strategies. Moreover, the lift performance of PETs and particularly ensembles of PETs has, to the best of our knowledge, never been analyzed in the context of customer churn prediction.

2.2 Ensemble Classification

Ensemble classification has been a popular field of research in recent years. Multiple studies have demonstrated the beneficial effect of combining many classification models into aggregated ensemble classifiers on classification accuracy (e.g., [18][19]). While several algorithms have been proposed in literature, many are inspired by two classical ensemble strategies: Bagging [11] and Boosting [14]. In Bagging (bootstrap aggregating), each member classifier in the ensemble is trained on a bootstrap sample (i.e., a random sample taken with replacement and with the same size) of the original training data. Member outputs are aggregated using majority voting: instances are assigned the class that is most frequently assigned by

the ensemble members [11]. Bagging can introduce a significance improvement in accuracy as a result of a reduction of variance versus individual decision trees. The most well-known boosting algorithm is AdaBoost [14]. In AdaBoost, instances that are mislabeled receive higher weight during consecutive training iterations and hence, the classifier is forced to concentrate on hard-to-predict instances. A related method to Bagging is the Random Subspace Method (RSM, [12]), also known as attribute bagging [20]. In RSM, a random feature subset is sampled for the training of an ensemble member. Finally, Bagging and RSM are combined in the SubBag algorithm, proposed in [13].

An important element of any ensemble classifier algorithm is the fusion rule, used to aggregate ensemble member's outputs to ensemble predictions. A categorization is often made between fusion rules for label outputs and fusion rules for continuous outputs [16]. A well-studied and simple combiner is plurality voting, as mentioned earlier in the case of Bagging. Other algorithms, such as RSM implement average aggregation, which takes the average of the ensemble members' outputs. For both methods, weights can be assigned to ensemble members which score higher on a performance metric of choice to obtain weighted majority voting or weighted average aggregation [16]. Training error rates are often used as weights. In this study, top p -th percentile lift and derivations are introduced as weights for weighted average aggregation fusion rules.

2.3 Lift

In the context of churn prediction, lift focuses on the segment of customers with the highest risk to the company, i.e. customers with the highest probability to churn. The definition of lift depends upon the percentage of riskiest customers one is considering for a retention campaign. Suppose that a company is interested in the top p -th percentile of most likely churners, based on predicted churn probabilities. The top p -th percentile lift then equals the ratio of the proportion of churners in the top p -th percentile of ordered posterior churn probabilities, $\pi_{p\%}$, to the churn rate in the total customer population, π ; *top p -th percentile lift* = $\frac{\pi_{p\%}}{\pi}$. As the proportion of customers that a company is able and willing to target depends on the specific content, the experiments will calculate lift performance for different percentiles. The concept of directly optimizing the lift measure is similar to the one proposed in [21]. The authors maximize the number of purchases at a given mailing depth (used as a constraint) for database marketing.

3 Experimental Validation

3.1 Data

To investigate the suitability of probability estimation trees as base classifiers in ensemble classifiers for increasing lift in churn prediction, this study considers five real-life churn data sets from different business contexts and for different

Table 1. Data set description

| <i>Data set</i> | <i>Instances</i> | <i>Features</i> | <i>Minority class percentage</i> |
|--------------------------|------------------|-----------------|----------------------------------|
| <i>Bank1</i> | 23,562 | 236 | 3.52 |
| <i>Bank2</i> | 42,783 | 164 | 11.14 |
| <i>Supermarket chain</i> | 32,371 | 46 | 25.15 |
| <i>DIY chain</i> | 3,827 | 15 | 28.14 |
| <i>Bank3</i> | 20,456 | 137 | 5.99 |

products or services. For reasons of confidentiality, company names are not disclosed. The characteristics of these data sets are provided in Table 1.

As shown in Table 1, churn data sets are typically characterized by relative high numbers of features and instances. An exception is the *DIY chain* data set. A second issue is the class imbalance of the data. Churn is usually a rare event [30]. This is particularly a problem for the *Bank1*, *Bank2* and *Bank3* data sets. Many techniques have been proposed to deal with class imbalance in churn prediction [7, 8]. In [22], the effect of class imbalance on a number of performance metrics is analyzed for probability estimation trees. This study indicates the importance of an appropriate treatment for the problem of class imbalance for the quality of probability estimates of PETs. While the authors suggest a wrapper method to determine an optimal sampling level of undersampling, in this study, majority class instances in the training data sets are undersampled in order to obtain balanced class distributions. In particular, the training data set for a classifier is composed of all instances belonging to the minority class and a random sample of majority class instances with a size equal to the number of minority class instances.

3.2 Experimental Settings

In this study, we consider four ensemble classifiers: Bagging, RSM, SubBag and AdaBoost. C4.4 and C4.5 base classifiers are implemented using the J48 classifier, which is a C4.5 implementation available in WEKA [23]. C4.4 is implemented as in [9]. Bagging, RSM, SubBag, AdaBoost and the proposed variations are implemented in MATLAB. Ensemble sizes are set to 100 constituent ensemble members. One final parameter is the random feature subset size for RSM and SubBag. This parameter is set equal to 75 % of the number of features in the respective data sets, as suggested in [13]. The default combination rule of Bagging and SubBag (i.e., majority voting) is replaced by average aggregation, as class predictions are not suited for an evaluation in terms of a rank-based measure such as lift.

A first comparison involves the use of C4.4 PETs versus standard C4.5 classification trees as base classifiers in ensemble classifiers. In a second comparison, the influence of the introduction of alternative fusion rules based on top p -th percentile lift is investigated. Throughout the comparisons, different lift definitions are considered, with p ranging from 50 to 5. More specifically, $p \in \{50, 40, 30, 20, 10, 5, p_r\}$ where p_r is the (rounded) actual churn rate as observed in the training data, which is provided in Table 1. This additional percentile

represents the often-used strategy of companies to target as many potential churners as they expect to emerge based on past experience.

Weighted average aggregation is applied using as weights: (i) top p -th percentile lift, further referred to as $lift_p$, (ii) $lift_p - 1$, and (iii) rescaled $lift_p$, defined as $\frac{lift_p - min_lift_p}{min_lift_p}$ where min_lift_p is the minimum over the lift values as calculated for the set of base classifiers. All weights are based on lift performance of the individual base classifiers on the training data set. Alternative (ii) is inspired by fact that a lift of 1 implies a model not outperforming random guessing. Only if the classifier outperforms random guessing is $(lift_p - 1)$ a positive weight for the respective classifier. In alternative (iii), the minimal observed lift is set as a minimum threshold for classifiers to receive a positive weight.

Reported results are averaged over a 5x2-fold cross-validation. Within a 2-fold cross-validation, the training set is randomly split into two parts; the first part is used for model training, while the second part is used for model validation and vice versa.

3.3 Results

Results are reported as counts of wins, losses and ties based on paired t-tests with significance level $\alpha = 0.05$. When a reference algorithm performs significantly better (worse) in terms of lift performance, a win (loss) is registered, while equal lift performance between a reference and a benchmark algorithm results in a tie. Tables 2 to 5 provide wins-losses-ties counts for variations based on Bagging, RSM, SubBag and AdaBoost. Tables 6 to 9 provide average performance ranks for the ensemble variations.

Table 2. Experimental results for Bagging: wins-losses-ties

| Algorithm | Bagging (C4.5 + averaging) | Bagging (C4.5 + lift weights) | Bagging (C4.5 + (lift-1) weights) | Bagging (C4.5 + rescaled lift weights) | Bagging (C4.4 + averaging) | Bagging (C4.4 + lift weights) | Bagging (C4.4 + (lift-1) weights) | Bagging (C4.4 + rescaled lift weights) |
|----------------------------------------|----------------------------------|-------------------------------------|--------------------------------------------|-------------------------------------------------|----------------------------------|-------------------------------------|--------------------------------------------|-------------------------------------------------|
| Bagging (C4.5 + averaging) | - | 1/1/32 | 1/2/32 | 3/0/32 | 0/10/25 | 0/9/26 | 0/11/24 | 0/9/26 |
| Bagging (C4.5 + lift weights) | 1/1/32 | - | 2/1/32 | 3/0/32 | 0/9/26 | 0/9/26 | 0/10/25 | 0/9/26 |
| Bagging (C4.5 + (lift-1) weights) | 2/1/32 | 1/2/32 | - | 2/0/33 | 0/9/26 | 0/9/26 | 0/9/26 | 0/8/27 |
| Bagging (C4.5 + rescaled lift weights) | 0/3/32 | 0/3/32 | 0/2/33 | - | 0/11/24 | 0/10/25 | 0/10/25 | 0/9/26 |
| Bagging (C4.4 + averaging) | 10/0/25 | 9/0/26 | 9/0/26 | 11/0/24 | - | 0/0/35 | 0/0/35 | 5/1/29 |
| Bagging (C4.4 + lift weights) | 9/0/26 | 9/0/26 | 9/0/26 | 10/0/25 | 0/0/35 | - | 0/0/33 | 4/2/29 |
| Bagging (C4.4 + (lift-1) weights) | 11/0/24 | 10/0/25 | 9/0/26 | 10/0/25 | 0/0/35 | 0/0/33 | - | 3/2/30 |
| Bagging (C4.4 + rescaled lift weights) | 9/0/26 | 9/0/26 | 8/0/27 | 9/0/26 | 1/5/29 | 2/4/29 | 2/3/30 | - |

A first set of observations is derived for the Bagging variations. The wins, losses and ties counts in table 2 indicate how the lift performance of Bagging can be increased by replacing standard C4.5 decision trees with C4.4 PETs, which confirms findings in [12]. This is also confirmed in table 6, where there is a clear dominance of all C4.4-based variations. The influence of the weighted combinations rules based on lift is less clear. It appears that the proposed weighing schemes marginally improve lift performance for C4.5-based Bagging. However, for Bagging with C4.4 member classifiers, the alternative combination rules do not result in an additional increase of lift performance.

Table 3. Experimental results for RSM: wins-losses-ties

| Algorithm | RSM (C4.5) | RSM (C4.5 + lift weights) | RSM (C4.5 + (lift-1) weights) | RSM (C4.5 + rescaled lift weights) | RSM (C4.4) | RSM (C4.4 + lift weights) | RSM (C4.4 + (lift-1) weights) | RSM (C4.4 + rescaled lift weights) |
|------------------------------------|---------------|---------------------------------|----------------------------------------|---------------------------------------------|---------------|---------------------------------|----------------------------------------|---------------------------------------------|
| RSM (C4.5) | - | 0/8/27 | 0/14/21 | 0/14/21 | 0/2/33 | 0/2/33 | 0/4/31 | 0/4/31 |
| RSM (C4.5 + lift weights) | 8/0/27 | - | 1/7/27 | 1/6/28 | 0/1/34 | 0/0/35 | 0/1/34 | 0/1/34 |
| RSM (C4.5 + (lift-1) weights) | 14/0/21 | 7/1/27 | - | 3/0/32 | 2/0/33 | 0/1/34 | 0/1/34 | 0/1/34 |
| RSM (C4.5 + rescaled lift weights) | 15/0/20 | 10/1/24 | 7/3/25 | - | 4/1/30 | 1/2/32 | 0/1/34 | 0/2/33 |
| RSM (C4.4) | 2/0/33 | 1/0/34 | 0/2/33 | 1/1/33 | - | 1/15/19 | 0/14/21 | 0/15/20 |
| RSM (C4.4 + lift weights) | 2/0/33 | 0/0/35 | 1/0/34 | 2/0/33 | 15/1/19 | - | 0/10/25 | 1/12/22 |
| RSM (C4.4 + (lift-1) weights) | 4/0/31 | 1/0/34 | 1/0/34 | 1/0/34 | 14/0/21 | 10/0/25 | - | 4/1/29 |
| RSM (C4.4 + rescaled lift weights) | 5/0/30 | 3/0/32 | 2/0/33 | 2/0/33 | 15/0/20 | 13/1/21 | 9/0/26 | - |

Table 4. Experimental results for SubBag: wins-losses-ties

| Algorithm | SubBag (C4.5 + averaging) | SubBag (C4.5 + lift weights) | SubBag (C4.5 + (lift-1) weights) | SubBag (C4.5 + rescaled lift weights) | SubBag (C4.4 + averaging) | SubBag (C4.4 + lift weights) | SubBag (C4.4 + (lift-1) weights) | SubBag (C4.4 + rescaled lift weights) |
|---------------------------------------|---------------------------------|------------------------------------|-------------------------------------------|------------------------------------------------|---------------------------------|------------------------------------|-------------------------------------------|------------------------------------------------|
| SubBag (C4.5 + averaging) | - | 1/4/30 | 2/5/28 | 2/4/29 | 2/0/33 | 2/3/30 | 4/3/28 | 3/7/25 |
| SubBag (C4.5 + lift weights) | 4/1/30 | - | 1/6/28 | 3/4/28 | 2/0/33 | 2/1/32 | 3/4/28 | 3/4/28 |
| SubBag (C4.5 + (lift-1) weights) | 5/2/28 | 6/1/28 | - | 1/2/32 | 2/0/33 | 2/0/33 | 2/3/30 | 2/2/31 |
| SubBag (C4.5 + rescaled lift weights) | 4/2/29 | 4/3/28 | 2/1/32 | - | 1/1/33 | 0/1/34 | 1/2/32 | 1/3/31 |
| SubBag (C4.4 + averaging) | 0/2/33 | 0/2/33 | 0/2/33 | 1/1/33 | - | 0/6/29 | 0/6/29 | 0/8/27 |
| SubBag (C4.4 + lift weights) | 3/2/30 | 1/2/32 | 0/2/33 | 1/0/34 | 6/0/29 | - | 1/4/30 | 1/7/27 |
| SubBag (C4.4 + (lift-1) weights) | 3/4/28 | 4/3/28 | 3/2/30 | 2/1/32 | 6/0/29 | 4/1/30 | - | 1/6/28 |
| SubBag (C4.4 + rescaled lift weights) | 7/3/25 | 4/3/28 | 2/2/31 | 3/1/31 | 8/0/27 | 7/1/27 | 6/1/28 | - |

Table 5. Experimental results for AdaBoost: wins-losses-ties

| Algorithm | AdaBoost (C4.5) | AdaBoost (C4.5 + lift weights) | AdaBoost (C4.5 + (lift-1) weights) | AdaBoost (C4.5 + rescaled lift weights) | AdaBoost (C4.4) | AdaBoost (C4.4 + lift weights) | AdaBoost (C4.4 + (lift-1) weights) | AdaBoost (C4.4 + rescaled lift weights) |
|-----------------------------------------|--------------------|--------------------------------------|---------------------------------------------|--------------------------------------------------|--------------------|--------------------------------------|---------------------------------------------|--------------------------------------------------|
| AdaBoost (C4.5) | - | 5/5/25 | 11/1/23 | 11/1/23 | 3/23/9 | 0/21/14 | 0/21/14 | 0/21/14 |
| AdaBoost (C4.5 + lift weights) | 5/5/25 | - | 18/0/17 | 21/0/14 | 4/24/7 | 1/24/10 | 2/24/9 | 1/24/10 |
| AdaBoost (C4.5 + (lift-1) weights) | 1/11/23 | 0/18/17 | - | 10/0/24 | 1/25/9 | 0/24/11 | 0/24/11 | 0/25/10 |
| AdaBoost (C4.5 + rescaled lift weights) | 1/11/23 | 0/21/14 | 0/10/24 | - | 1/25/9 | 0/24/11 | 0/24/11 | 0/25/10 |
| AdaBoost (C4.4) | 23/3/9 | 24/4/7 | 25/1/9 | 25/1/9 | - | 3/12/20 | 3/11/21 | 0/4/31 |
| AdaBoost (C4.4 + lift weights) | 21/0/14 | 24/1/10 | 24/0/11 | 24/0/11 | 12/3/20 | - | 1/0/34 | 5/0/30 |
| AdaBoost (C4.4 + (lift-1) weights) | 21/0/14 | 24/2/9 | 24/0/11 | 24/0/11 | 11/3/21 | 0/1/34 | - | 5/1/29 |
| AdaBoost (C4.4 + rescaled lift weights) | 21/0/14 | 24/1/10 | 25/0/10 | 25/0/10 | 4/0/31 | 0/5/30 | 1/5/29 | - |

Table 6. Experimental results for Bagging: average ranks

| Algorithm | Ranking |
|----------------------------------------|---------|
| Bagging (C4.4 + averaging) | 3.4286 |
| Bagging (C4.4 + rescaled lift weights) | 3.4286 |
| Bagging (C4.4 + (lift-1) weights) | 3.4714 |
| Bagging (C4.4 + lift weights) | 3.8143 |
| Bagging (C4.5 + (lift-1) weights) | 5.3143 |
| Bagging (C4.5 + rescaled lift weights) | 5.5143 |
| Bagging (C4.5 + lift weights) | 5.5286 |
| Bagging (C4.5 + averaging) | 5.6143 |

Tables 3 and 7 present results for the Random Subspace Method. Here different results are found. The introduction of C4.4 as base classifier in RSM only very slightly improves performance over RSM with C4.5 base classifiers. However, fusion rules based on top p -th percentile lift invoke substantial improvements of lift performance over average aggregation for RSM, regardless of the nature of its base classifiers. The best results are observed for rescaled lift and ($lift - 1$) weights.

Table 7. Experimental results for RSM: average ranks

| <i>Algorithm</i> | <i>Ranking</i> |
|-------------------------------------------|----------------|
| <i>RSM (C4.4 + (lift-1) weights)</i> | 3.2143 |
| <i>RSM (C4.5 + (lift-1) weights)</i> | 3.4286 |
| <i>RSM (C4.4 + rescaled lift weights)</i> | 3.7857 |
| <i>RSM (C4.5 + rescaled lift weights)</i> | 3.8857 |
| <i>RSM (C4.4 + lift weights)</i> | 4.6000 |
| <i>RSM (C4.5 + lift weights)</i> | 4.9000 |
| <i>RSM (C4.4)</i> | 6.0571 |
| <i>RSM (C4.5)</i> | 6.1286 |

Table 8. Experimental results for SubBag: average ranks

| <i>Algorithm</i> | <i>Ranking</i> |
|----------------------------------------------|----------------|
| <i>SubBag (C4.5 + rescaled lift weights)</i> | 3.8429 |
| <i>SubBag (C4.5 + (lift-1) weights)</i> | 4.0429 |
| <i>SubBag (C4.5 + lift weights)</i> | 4.1857 |
| <i>SubBag (C4.4 + rescaled lift weights)</i> | 4.4571 |
| <i>SubBag (C4.4 + (lift-1) weights)</i> | 4.5857 |
| <i>SubBag (C4.5 + averaging)</i> | 4.6143 |
| <i>SubBag (C4.4 + lift weights)</i> | 5.0286 |
| <i>SubBag (C4.4 + averaging)</i> | 5.2429 |

Table 9. Experimental results for AdaBoost: average ranks

| <i>Algorithm</i> | <i>Ranking</i> |
|------------------------------------------------|----------------|
| <i>AdaBoost (C4.4 + lift weights)</i> | 3.0857 |
| <i>AdaBoost (C4.4 + rescaled lift weights)</i> | 3.1143 |
| <i>AdaBoost (C4.4 + (lift-1) weights)</i> | 3.2857 |
| <i>AdaBoost (C4.4)</i> | 4.0000 |
| <i>AdaBoost (C4.5 + lift weights)</i> | 4.7857 |
| <i>AdaBoost (C4.5)</i> | 5.3143 |
| <i>AdaBoost (C4.5 + (lift-1) weights)</i> | 6.0143 |
| <i>AdaBoost (C4.5 + rescaled lift weights)</i> | 6.4000 |

For SubBag, the implementation of C4.4 member classifiers and alternative fusion rules has no clear impact upon lift. While the average rankings in table 8 indicate an advantage for C4.5-based SubBag with lift-weighted fusion rules, the wins, losses and ties counts suggest that the differences are small.

Finally, consider the experimental results for AdaBoost in tables 5 and 9. Here, both modified base learners and adapted fusion rules contribute to an improvement of lift performance. The introduction of C4.4 base learners generates a marked improvement over using C4.5 base classifiers. The use of lift-based fusion rules invokes a further increase. The best performance is observed for regular lift weights.

4 Conclusion

Customer retention and churn prediction are important elements of Customer Relationship Management (CRM) strategies. An often-used evaluation metric for churn-prediction models is lift, which measures the degree to which the

model is better in identifying the customers most likely to churn, over random guessing. This study investigates strategies to improve lift performance of four well-known ensemble algorithms. The first is the adoption of C4.4 probability estimation trees (PETs) as base classifiers, instead of regular C4.5 classification trees. The second strategy involves replacing standard fusion rules for ensemble members' outputs by weighted average aggregation, using three alternative lift-based weight sets. Both strategies are applied to four well-known ensemble algorithms: Bagging, the Random Subspace Method (RSM), SubBag and AdaBoost. Experiments on five real-life churn prediction data sets are conducted to compare C4.5 and C4.4 trees as base classifiers, and original versus the proposed fusion rules. The results indicate variation in the effect of the proposed strategies on lift performance depending on the nature of the ensemble algorithm: (i) Bagging greatly benefits from adopting C4.4 base classifiers using average aggregation as a fusion rule, while lift-based weighted averaging does not substantially improve lift performance; (ii) weighted average aggregation is a viable strategy to increase lift performance of RSM, while it does not benefit from adopting C4.4 trees as base classifiers; (iii) SubBag does not notably benefit from either of both strategies, and (iv) the lift performance of AdaBoost can be substantially improved by implementing both C4.4 base classifiers and weighted average aggregation based on lift.

Certain limitations to this study can be identified. While the results clearly reveal a number of trends, the authors acknowledge that further experiments are needed to allow for statistically proven generalizations. Further, no comparisons are made between the proposed methods and classification algorithms that are designed to optimize ranking performance such as AUC. Future work includes the adoption of different PET algorithms and alternative ensemble strategies, such as the recently proposed Rotation Forests [26] and application of error decomposition and other techniques to gain insight in the accuracy-diversity trade-off to gain more insight in the specific conditions that need to be fulfilled to successfully ensemble PETs for increasing lift performance.

Acknowledgements

The authors thank the reviewers for reviewing the paper and Ghent University for funding the PhD project of Koen W. De Bock.

References

1. Reinartz, W., Kumar, V.: The mismanagement of customer loyalty. *Harvard Bus. Rev.* 80, 86–94 (2002)
2. Shaw, M.J., Subramaniam, C., Tan, G.W., Welge, M.E.: Knowledge management and data mining for marketing. *Decis. Support Syst.* 31, 127–137 (2001)
3. Kim, Y.S.: Toward a successful CRM: variable selection, sampling, and ensemble. *Decis. Support Syst.* 41, 542–553 (2006)

4. Larivière, B., Van den Poel, D.: Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Syst. Appl.* 29, 472–484 (2005)
5. Jinbo, S., Xiu, L., Wenhua, L.: The application of AdaBoost in customer churn prediction. In: *Proceedings of 2007 International Conference on Service Systems and Service Management (ICSSSM 2007)*, pp. 513–518 (2007)
6. Gladly, N., Baesens, B., Croux, C.: Modeling churn using customer lifetime value. *Eur. J. Oper. Res.* 197, 402–411 (2009)
7. Lemmens, A., Croux, C.: Bagging and boosting classification trees to predict churn. *J. Marketing Res.* 43, 276–286 (2006)
8. Burez, J., Van den Poel, D.: Handling class imbalance in customer churn prediction. *Expert Syst. Appl.* 36, 4626–4636 (2009)
9. Provost, F., Domingos, P.: Tree induction for probability-based ranking. *Mach. Learn.* 52, 199–215 (2003)
10. Quinlan, R.: *C4.5: Programs for Machine Learning*. Morgan Kaufman Publishers, San Mateo (1993)
11. Breiman, L.: Bagging predictors. *Mach. Learn.* 24, 123–140 (1996)
12. Ho, T.K.: The random subspace method for constructing decision forests. *IEEE T. Pattern Anal.* 20, 832–844 (1998)
13. Panov, P., Dzeroski, S.: Combining bagging and random subspaces to create better ensembles. In: Berthold, M.R., Shawe-Taylor, J., Lavrac, N. (eds.) *IDA 2007*. LNCS, vol. 4723, pp. 86–94. Springer, Heidelberg (2007)
14. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* 55, 119–139 (1997)
15. Clemençon, S., Vayatis, N.: Tree-Based Ranking Methods. *IEEE T. Inform. Theory* 55, 4316–4336 (2009)
16. Kuncheva, L.I.: *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, Hoboken (2004)
17. Provost, F., Fawcett, T., Kohavi, R.: The Case against Accuracy Estimation for Comparing Induction Algorithms. In: Shavlik, J. (ed.) *15th International Conference on Machine Learning (ICML-1998)*, pp. 445–453. Morgan Kaufman, San Francisco (2000)
18. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.* 36, 105–139 (1999)
19. Rodríguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. *IEEE T. Pattern Anal.* 28, 1619–1630 (2006)
20. Bryll, R., Gutierrez-Osuna, R., Quek, F.: Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recogn.* 36, 1291–1302 (2003)
21. Prinzie, A., Van den Poel, D.: Constrained optimization of data-mining problems to improve model performance: A direct-marketing application. *Expert Syst. Appl.* 29, 630–640 (2005)
22. Cieslak, D., Chawla, N.: Analyzing PETs on imbalanced datasets when training and testing class distributions differ. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) *PAKDD 2008*. LNCS (LNAI), vol. 5012, pp. 519–526. Springer, Heidelberg (2008)
23. Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: *The WEKA Data Mining Software: An Update*. *SIGKDD Explorations* 1 (2009)

Evolving Ensembles of Feature Subsets towards Optimal Feature Selection for Unsupervised and Semi-supervised Clustering

Mihaela Elena Breaban

Faculty of Computer Science, Al. I. Cuza University, Iasi, Romania
pmihaela@infoiasi.ro

Abstract. The work in unsupervised learning centered on clustering has been extended with new paradigms to address the demands raised by real-world problems. In this regard, unsupervised feature selection has been proposed to remove noisy attributes that could mislead the clustering procedures. Additionally, semi-supervision has been integrated within existing paradigms because some background information usually exist in form of a reduced number of similarity/dissimilarity constraints. In this context, the current paper investigates a method to perform simultaneously feature selection and clustering. The benefits of a semi-supervised approach making use of reduced external information are highlighted against an unsupervised approach. The method makes use of an ensemble of near-optimal feature subsets delivered by a multi-modal genetic algorithm in order to quantify the relative importance of each feature to clustering.

Keywords: unsupervised and semi-supervised learning, clustering, feature selection, feature ranking, ensemble learning.

1 Introduction

Classification and clustering are two problems intensively studied in machine learning. Classification aims at assigning new data items to existing groupings. Clustering is the problem of identifying natural or interesting groupings in data. Although similar at first sight, they belong to two distinct paradigms: supervised versus unsupervised learning.

Feature selection (FS) is a problem of great interest for both scenarios - classification and clustering - with the aim of improving the performance of the corresponding machine learning techniques. Feature ranking is a relaxation of FS: the features are ranked based on their relevance to the problem under investigation. With regard to clustering, fewer approaches exist in literature due to the difficulties raised by the unsupervised nature of the problem; most of them offer feature rankings because the optimal number of features to be selected is hard to be determined in the unsupervised scenario.

The current work investigates an extension of a feature ranking technique we have recently proposed in the context of unsupervised clustering [1]. The method

is based on evolving an ensemble of feature subsets which serve further for feature ranking. The algorithm is extended here to return the optimal subset of features for clustering, overcoming the initial drawback of fixed cardinality imposed over the feature subspace. As a result, the extended method is able to return the optimum number of features in a completely unsupervised scenario. Additionally, an extension is proposed to deal with the semi-supervised version of clustering, namely supervised information is incorporated in form of similarity/dissimilarity pairwise constraints.

The paper is structured as follows. Section 2 presents succinctly existing approaches to feature selection and feature ranking for unsupervised and semi-supervised clustering. Section 3 revisits the method we proposed previously for feature ranking [1] and describes an extension for the semi-supervised case. Section 4 extends the method towards a complete feature selection procedure and Section 5 evaluates the performance of the method on complex synthetic data sets. The paper concludes with a short discussion in Section 6.

2 Related Work on Feature Selection and Feature Ranking

Feature selection (FS) generally aims at reducing the representation of the data in order to lower the computational cost of further analysis. To this goal, filter methods were designed which try to remove redundant features in a pre-processing step.

With regard to the two intensively-studied problems in machine learning - classification and clustering - FS plays different roles. In classification FS aims to identify the features which predict with the highest accuracy the class labels. In clustering FS aims to identify the features which lead to well-defined groupings.

Feature ranking is a generalization of FS: the features are ordered in accordance with their relative contribution to the goals expressed previously.

2.1 The Unsupervised Scenario

In the unsupervised scenario for clustering no information is available with regard to the number of clusters nor with regard to the assignment of some particular data instances.

Filter approaches to FS aim at quantifying the merit of each feature ignoring the subsequent method of analysis. Two strategies are used to compute the merit of each feature: one that aims at removing redundant features and one that scores the relevance of features. Redundancy-based approaches hold that mutually-dependent features should be discarded. In this regard, clustering on features is proposed selecting further one representative feature per class [4]. On the contrary, there exist approaches in the second category [11,14] that compute relevance accepting that relevant features are highly dependent on the clusters structure and therefore, they are pairwise dependent; pairwise dependence scores are computed using mutual information and mutual prediction [14]. Other

approaches rank the features accordingly to their variances or accordingly to their contribution to the entropy calculated on a leave-one-out basis [153].

Unlike most of the filter approaches, wrapper methods evaluate feature subsets and not simple features. These approaches perform better since the evaluation is based on the exploratory analysis method employed for data analysis. However, wrapper approaches have two drawbacks: high computational time, and bias. The high computational time is due to the evaluation procedure which consists in running a full clustering algorithm. The bias is due to the objective function used to evaluate different partitions. In an unsupervised framework, the objective function which guides the search for good partitions induces some biases on the number of clusters and the size of the feature subspace. Regarding the number of clusters, several unsupervised clustering criteria were proposed to deal with the unsupervised clustering problem (i.e. Silhouette Width and Davis-Bouldin Index). However, all objective functions are based on computing some distance function for every pair of data items; the dimensionality influences the distribution of the distances between data items and thus induces a bias on the size of the feature space.

In order to reduce the bias with regard to the number of features, a few strategies were proposed. Dy and Brodley [5] introduce the cross-projection normalization: given two feature subsets, the best partition is determined for each feature subspace and the resulting partitions are each evaluated in the other subspace. The cross-projection normalization is used in a greedy scenario (sequential forward search); because it is not transitive, its use in global search techniques is inappropriate. However, this drawback is alleviated in [1] by using a steady-state genetic algorithm which implements a crowding scheme at replacement encouraging the competition among pairs of relatively similar feature subsets. The cross-projection normalization is thus studied in a larger context and its performance proved to be highly dependent on the unsupervised clustering criteria used: i.e. Silhouette Width [13] which is one of the best performers in unsupervised clustering behaved worst under this normalization. In our opinion, the cross-projection normalization does not offer a feasible solution to the feature cardinality bias: the results regarding feature selection show that part of the relevant features are eliminated and, furthermore, irrelevant features are selected as relevant. This suggests that the bias towards small features subspaces is not completely removed and that the cross-projection misleads the search algorithm.

Multi-objective optimization algorithms are a more straightforward way to deal with biases: the bias introduced in the primary objective function is counterbalanced by other objective functions. A more extensive study on the use of multi-objective optimization for unsupervised feature selection is carried out in [7]: some drawbacks of the existing methods are outlined and several objective functions are thoroughly tested on a complex synthetic benchmark.

More recently, ensemble unsupervised feature ranking and selection were proposed. In [9] clustering is performed on random subsets of features and each feature is ranked through analyzing the correlations between the features and the clustering solution. Based on the ensemble of feature rankings, one consensus ranking is

constructed. Even if this ensemble feature ranking method is considered to work unsupervised, some degree of supervision is introduced in the study: in order to deliver partitions in feature subspaces, k-Means is run with the known number of clusters. In [6] an ensemble of feature rankings is obtained using the Laplacian score for random feature subsets and a subset of the best features with respect to the different rankings is obtained by assuming that the distribution of the irrelevant features at each of the remaining k th rank is uniform.

2.2 The Semi-supervised Scenario

In the semi-supervised scenario for clustering, external information is introduced in the form of a reduced number of pairwise constraints: similarity constraints indicate pairs of data items which must share the same cluster and dissimilarity constraints indicate pairs of data items which must be put in different clusters. The number of clusters is still unknown; however, some information with regard to the minimum number of clusters allowed can be inferred from the constraints.

The semi-supervised problem stands as a junction for supervised learning and unsupervised learning. Therefore, two wrapper scenarios are proposed in literature: 1) classifiers are extended to incorporate unlabeled data and 2) clustering methods are modified to benefit from guidance provided by the labeled data.

In the first category, Ren et.al. [12] learn a classifier on the reduced labeled data and extend it at each iteration introducing randomly-selected unlabeled data; implicitly, new features are added iteratively in a forward-search manner.

In the second category, Handl and Knowles extend their multi-objective algorithm proposed for unsupervised feature selection [8]. The Adjusted Rand Index (ARI) [10] is introduced to measure the consistency with the given constraints or class labels as a third objective or in a linear and non-linear combination with the unsupervised clustering criterion. The solution recording the highest consistency reflected by the ARI score is reported from the final Pareto front.

3 Constructing an Ensemble of Near-Optimal Feature Subsets

In [1] a multi-modal genetic algorithm is used to derive an ensemble of near-optimal feature subsets for unsupervised clustering. The wrapper paradigm is employed. The bias with regard to the cardinality of the feature subsets introduced by the unsupervised clustering criteria (as explained in Section 2.1) impedes us to select the optimal feature subset. For this reason a fixed cardinality is imposed for all feature subsets and the ensemble of solutions in the final generation of the genetic algorithm is used to derive feature weights which lead further to an optimal ranking. Experimental results showed that the method is feasible in the context of data sets with a large number of noisy features. However, its performance is dependant on the cardinality of the candidate feature subsets imposed within population. The current paper alleviates this drawback and proposes a complete feature selection algorithm.

Subsection 3.1 revisits the genetic algorithm in [11] and Subsection 3.2 proposes an extension for the semi-supervised case.

3.1 The Unsupervised Scenario

The use of a multi-modal algorithm to search for optimal feature subsets is highly justified by the multi-modal nature of the problem: different feature subspaces may lead to different meaningful partitions of the original data. However, the reason for using such an approach in the current paper is different: the diversity in population maintained by such an algorithm will guarantee an uniform distribution of the irrelevant features. This premise of uniform distribution of the irrelevant features has been already used in the ensemble FS method presented in [6] but the framework we propose here is totally different.

The algorithm we use is the Multi-Niche Crowding genetic algorithm (MNC-GA) [16]. It is employed for several reasons: it maintains diversity throughout the search and maintains stable sub-populations within different niches of the search space. Also, by implementing a steady-state scheme it allowed us to study in previous work the cross-projection normalization [11].

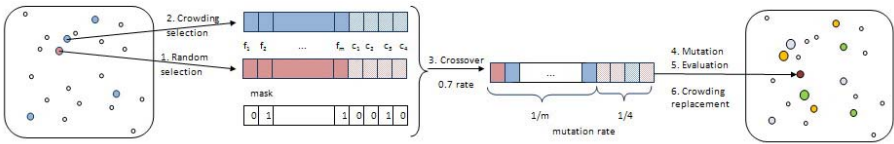


Fig. 1. One iteration in the MNC-GA algorithm for FS

Figure 1 illustrates one iteration of the MNC-GA algorithm for feature selection. A chromosome in the algorithm encodes a feature subspace and the number of clusters of the partition to be derived in that feature subspace. The evaluation of such a chromosome necessitates a clustering algorithm: the k-Means algorithm is used to return the near-optimal partition in the encoded feature subspace with the number of clusters specified in the encoding. The evaluation of a feature subset is thus reduced to the evaluation of the derived partition. Unsupervised clustering criteria able to evaluate partitions with various numbers of features and built over feature subspaces of different cardinalities are required. Widely used unsupervised clustering criteria like Silhouette [13] and Davis-Bouldin [2] would favor feature subsets of lower cardinality and would direct the search towards the feature subset of minimum allowed cardinality. We proposed a new unsupervised clustering criterion which reduces part of this bias by penalizing small numbers of features m :

$$Crit = \left(\frac{1}{1 + \frac{W}{B}} \cdot \frac{m}{m + 0.5} \right)^{\log_2(k+1)+1} \quad (1)$$

where $W = \sum_{i=1}^k \sum_{d \in C_i} s(c_i, d)$ is the within-cluster inertia computed as the sum of the distances between all data items d and their cluster centers c_i ;

$B = \sum_{i=1}^k |C_i| \cdot s(c_i, g)$ is the between-cluster inertia computed as the sum of the distances between the cluster centers c_i and the center of the entire data set g weighted with the size of each cluster $|C_i|$.

Experiments on synthetic data sets with the criterion in equation 1) suggested that it is a brave competitor to the Silhouette Width criterion, working at reduced computational costs; it outperformed the Davis-Bouldin Index in our experiments [1]. It still records a bias towards lower cardinalities of the feature space but not as pronounced as the bias introduced by the other unsupervised clustering criteria.

To deal with the bias introduced by the unsupervised clustering criteria, limits are imposed to the number of features selected within a chromosome.

3.2 The Semi-supervised Scenario

We investigate two scenarios to introduce external information in the previous unsupervised approach. In a first scenario, the fitness function is modified to reflect the consistency of the partition with the labeled data: the product between the unsupervised clustering criterion in equation 1) and the ARI [10] score involving the labeled data is used. In the second scenario we force all partitions to satisfy the given constraints by employing constrained KMeans [17] as clustering procedure.

4 Feature Selection

In [1] feature rankings are derived based on the distribution of the features encoded in the final population of the algorithm described previously. The success of this ranking scheme is based on two premises: 1) uniform distribution of the irrelevant features and 2) high frequency of the relevant features in the final generation of the genetic algorithm. With regard to the roles of the genetic operators, the mutation is responsible for diversity which supports the first premise while the crossover operator propagates in population the best characteristics, supporting the second premise.

Because of the bias with regard to the number of features introduced by the fitness function, the chromosomes encode feature subspaces of fixed cardinality which is a parameter of the algorithm. This is an important drawback of the algorithm: if the number of relevant features is much smaller compared to the cardinality imposed, the relevant features get suffocated and the partition resulted does not reflect the distribution of values across the relevant features. From this point of view we anticipate that the ensemble method proposed in [9] which is based on measuring the correlation between the variables and the clustering solution suffers from the same drawback.

To overcome this drawback and to develop a method able to go further and perform feature selection we propose to vary the cardinality of the feature subspaces along the run of the genetic algorithm. The main decision factors involved are the variance of the fitness in population and the distribution of features in population reported to the cardinality of the encoded feature subspaces.

Regarding the dynamic of the fitness variance in population along the run, a behavior typical to genetic algorithms is recorded. A small variance in the first iteration is due to sub-optimal solutions. Once good schemata are retrieved, the variance increases due to the presence of a small number of high-fitness chromosomes. Then, the variance in fitness decreases as the population tends to converge. When the variance in fitness is smaller than the variance recorded in the first iteration and it remains unmodified for several iterations, the multi-modal genetic algorithm reaches convergence. It is worth noticing that we do not condition convergence to null variance, for several reasons. First of all, the multi-modal genetic algorithm is supposed to converge to multiple optima in the search space which signify different fitness values in the final iteration. Secondly, the fitness of the chromosomes is computed based on the partitions generated with k-Means; therefore, two identical chromosomes encoding the same feature subspace and equal numbers of clusters, could have been assigned different fitness values because of slightly different partitions generated as result of different initialization of the clustering algorithm.

When the conditions required for convergence are fulfilled, the distribution of selected features in population is computed. A heuristic step is employed at this stage: a feature is considered *relevant* if its frequency in population exceeds 50% (more than half of the chromosomes in population selects it). On this basis, the number of *relevant* features is computed; if it is smaller than the cardinality of the feature subspace imposed to chromosomes, the cardinality is decremented by 1 and the algorithm is restarted. To benefit from the information gathered throughout the search one new chromosome is constructed encoding the features marked as *relevant* and adding random chosen features to reach the cardinality imposed. To avoid the hitchhiking phenomenon which was shown to cause premature convergence in GAs, only the 25% best chromosomes are kept and the rest of the population is randomly generated, encouraging diversity.

When the conditions required for convergence are fulfilled, and the cardinality imposed to the feature subsets encoded by chromosomes does not exceed the number of features computed as relevant, the algorithm returns the features marked as relevant.

5 Experiments

In order to compare the results obtained by the new method with related work, the experiments are carried out on the artificial data sets created by Handl and Knowles [7]. A number of 40 data sets are used in our experiments, clustered into 4 groups (10 data sets per group) named *dd-kk*; *d* expresses the number of relevant features and takes the values {2,10} and *k* expresses the number of clusters and takes the values {4, 10}. For all data sets 100 Gaussian features were introduced as noise. Before applying the algorithm all data sets are standardized to have mean 0 and variance 1 for each feature. For the semi-supervised scenario, five data items are extracted randomly from each class and are used further as labeled data.

The parameters of the algorithm are set as follows. The population size was set to 50. In order to ensure diversity throughout the run in MNC-GA the size of the group at selection, the size of the group at replacement and the number of groups at replacement are all set to 10% of the population size. The number of features selected in each chromosome was set to 20 and then decreased along the run as explained in section 4; the choice of this specific value was made in order to be consistent with the experiments presented in [7] where the cardinality of the candidate feature subsets varies in the range 1-20.

The performance of our method is evaluated with regard to the quality of the partition obtained and with regard to the consistency between the feature subset returned and the relevant feature subset.

The partition reported in our experiments is obtained running k-Means with different numbers of clusters on the feature subset returned by our method. In the unsupervised case the best partition is extracted using the clustering criterion in equation 1). In the semi-supervised case, in the first scenario the product between ARI and the clustering criterion is used, while in the second scenario the constrained k-Means is used in conjunction with the clustering criterion. The Adjusted Rand Index (ARI) [10] is used to evaluate the partitions delivered by our method against the known true partition of the data set.

In order to judge the consistency between the returned feature subset and the known relevant feature subset, two measures from information retrieval are employed. Precision is defined as the number of relevant features identified divided by the total number of features returned and stands for Specificity. Recall is defined as the number of relevant features retrieved divided by the total number of existing relevant features and stands for Sensitivity. Also, their combination under the harmonic mean, known as F-measure, is reported.

As measure of time-complexity, the number of fitness evaluations required for a complete run of the algorithm is computed.

Table 1 presents the results for the unsupervised scenario as averages over 10 runs for each data set, 10 data sets per problem class.

Figure 2 (left) includes for comparison purposes the results presented in [7] obtained with the multi-objective genetic algorithm in a wrapper context and also in a filter scenario based on entropy. These results were obtained in a supervised manner from the Pareto front; a small decrease in performance is recorded if an automatic extraction procedure is involved, as shown in [7]. Figure 2 (right) presents the results obtained for semi-supervised feature selection.

Table 1. Results for unsupervised feature selection as averages over 10 runs for each data set: the ARI score and the number of clusters k for the best partition, the sensitivity and the specificity of the selected feature subspace

| Problem | ARI | k | sensitivity | specificity | F-measure | # evaluations |
|---------|--------|------|-------------|-------------|-----------|---------------|
| 2d-4c | 0.6623 | 3.98 | 0.89 | 0.93 | 0.90 | 12036 |
| 2d-10c | 0.70 | 8.78 | 0.97 | 0.99 | 0.98 | 11767 |
| 10d-4c | 0.9374 | 3.71 | 0.92 | 0.93 | 0.91 | 7887 |
| 10d-10c | 0.8055 | 8.16 | 0.93 | 0.99 | 0.95 | 8222 |

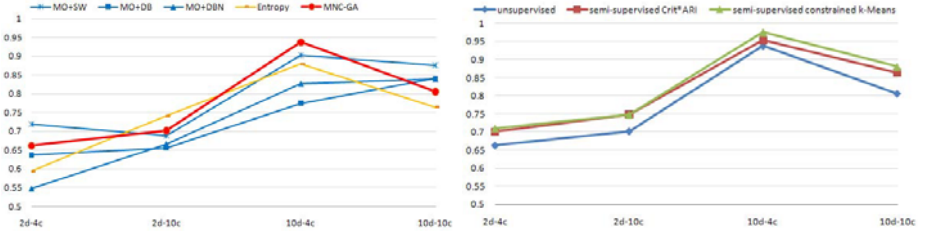


Fig. 2. ARI - comparative results. **Left:** the three lines denoted *MO-* correspond to the multi-objective algorithm investigated in [7] within a wrapper scenario with several clustering criteria used as the primary objective: Silhouette Width, Davies Bouldin and Davies-Bouldin normalized with respect to the number of features; *Entropy* corresponds to the multi-objective algorithm investigated in [7] within a filter scenario which is based on an entropy measure; *MNC-GA* corresponds to the method investigated in the current paper. **Right:** the unsupervised scenario and the two semi-supervised approaches.

The results in Table 1 show that the method is able to identify the relevant features and delivers high-quality partitions. The comparisons with the multi-objective algorithm, which is one of the few feasible solutions to unsupervised FS, show that the multi-modal approach behaves comparable.

Regarding the semi-supervised scenario, the gain in performance is evident compared to the unsupervised case, especially for the data sets with high numbers of clusters. The experiments show that constraining the partitions to satisfy the labeled data employing constrained k-Means generally hastens the retrieval of the relevant features and provides better results compared to the alternative approach. Additional experiments we have performed with higher numbers of labeled data items revealed that no significant improvements are obtained for the method which incorporates the supervised information in the fitness function. However, the method which makes use of Constrained k-Means continues to record performance improvements when increasing the number of labeled samples because the partitions are guaranteed to satisfy the provided labels.

6 Conclusion

Feature selection for unsupervised clustering is a problem of great interest in the current context of data mining. However, few feasible solutions exist in literature due to the difficulties raised by the unsupervised nature of the problem. Ensemble methods are gaining ground in this context. The current paper proposes a wrapper feature selection algorithm for data sets with large numbers of noisy features. The method makes use of an ensemble of near-optimal feature subsets evolved with a multi-modal genetic algorithm. Such an optimization algorithm is necessary to retrieve the relevant features in a large search space. At the same time, it is capable of maintaining high diversity in population, ensuring uniform distribution for the irrelevant features.

Acknowledgements

We would like to thank Julia Handl and Joshua Knowles for supplying us with the data sets investigated in the experimental section and with the results they obtained, making thus possible the reported comparisons with their extensive studies in unsupervised feature selection and clustering.

References

1. Breaban, M., Luchian, H.: Unsupervised feature weighting with multi-niche genetic algorithms. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation, July 2009, pp. 1163–1170. ACM, New York (2009)
2. Davies, D.L., Bouldin, D.W.: A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1(2), 224–227 (1979)
3. Domeniconi, C., Al-Razgan, M.: Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data* 2(4), 1–40 (2009)
4. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. John Wiley & Sons, Chichester (2001)
5. Dy, J., Brodley, C.: Feature selection for unsupervised learning. *Journal of Machine Learning Research* 5, 845–889 (2004)
6. Guerif, S.: Unsupervised variable selection: when random rankings sound as irrelevancy. *Journal of Machine Learning Research* 4, 163–177 (2008)
7. Handl, J., Knowles, J.: Feature subset selection in unsupervised learning via multiobjective optimization. *International Journal of Computational Intelligence Research* 2(3), 217–238 (2006)
8. Handl, J., Knowles, J.: Semi-supervised feature selection via multiobjective optimization. In: Proceedings of the International Joint Conference on Neural Networks, pp. 3319–3326 (2006)
9. Hong, Y., Kwong, S., Chang, Y., Ren, Q.: Consensus unsupervised feature ranking from multiple views. *Pattern Recognition Letters* 29, 595–602 (2008)
10. Hubert, A.: Comparing partitions. *Journal of Classification* 2, 193–198 (1985)
11. Talavera, L.: Feature selection as a preprocessing step for hierarchical clustering. In: Proceedings of the Sixteenth International Conference on Machine Learning, pp. 389–398. Morgan Kaufmann, San Francisco (1990)
12. Ren, J., Qiu, Z., Fan, W., Cheng, H., Yu, P.S.: Forward semi-supervised feature selection. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 970–976. Springer, Heidelberg (2008)
13. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20(1), 53–65 (1987)
14. Sndberg-madsen, N., Thomsen, C., Pena, J.M.: Unsupervised feature subset selection. In: Proceedings of the Workshop on Probabilistic Graphical Models for Classification (within ECML 2003) (2003)
15. Varshavsky, R., Gottlieb, A., Linial, M., Horn, D.: Novel unsupervised feature filtering of biological data. *Bioinformatics* 22(14), 507–513 (2006)
16. Vemuri, V., Cedeno, W.: Multi-niche crowding for multimodal search. In: *Practical Handbook of Genetic Algorithms: New Frontiers*, 2nd edn., Lance Chambers (1995)
17. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 577–584. Springer, Heidelberg (2001)

Building a New Classifier in an Ensemble Using Streaming Unlabeled Data

Mehmed Kantardzic, Joung Woo Ryu, and Chamila Walgampaya

CECS Department, Speed School of Engineering, University of Louisville,
Louisville, KY 40292, USA

mmkant01@louisville.edu, ryu0914@gmail.com,
ckwalg01@louisville.edu

Abstract. It is expensive and impractical to manually label all samples in real-world streaming data when the correct class is not available in real time. In this paper, we propose an ensemble method of determining which samples should be labeled from streaming unlabeled data and when they will be labeled according to changes in distribution of streaming unlabeled data. In particular, the labeling point in time is an important factor for building an efficient ensemble in practical aspects. In order to evaluate the performance of our ensemble method, we used synthetic streaming data with concept drift and the intrusion detection data from the KDD'99 Cup. We compared the results of the proposed method and those of the existing ensemble methods that periodically build new classifiers for an ensemble. In the synthetic streaming data, the proposed method produced average 14.1% higher classification accuracy, and the number of new classifiers reduced by average 12.6%. With the intrusion detection data, our method produced similar accuracy to existing methods but used only 0.007% of the labeled streaming data.

Keywords: Ensemble, Unlabeled data, Streaming data, Concept drift.

1 Introduction

Streaming data continuously flows in and out of a computer system until it shuts down. This means that streaming data is also a potentially infinite source. Today, streaming data is ubiquitous. Streaming-data mining (storing, analyzing, and visualizing such a continuous, infinite sequence of data) is a challenging task. In particular, classification techniques in the streaming-data mining can be applied to real-time decision support in business and industrial applications [1].

In order to generate a classification model, it is essential to label samples [7]. Most classification techniques for classifying streaming data assume availability of labels for all previous samples. The incremental learning methodology immediately requires the correct class of a new sample after classifying the new sample [4]. The ensemble methodology that periodically builds a new classifier for an ensemble can call for correct classes of accumulated samples when building a new classifier [3,6,7,9-12]. However, in online applications, such as intrusion detection and click fraud, labeling all previous streaming data may be costly and time-consuming.

In terms of the classification accuracy, Cozman et al. [2] and Kuncheva et al [5] analyzed the effect of unlabeled training data using Naïve Bayes rule. They demonstrated that if labeled training data correctly represents the underlying distribution of a problem, then unlabeled data is expected to improve upon classification error. However, if labeled training data is biased, using unlabeled data may do more harm than good. Therefore, it is another issue of streaming data classification to use unlabeled samples for classifying streaming data with change in distributions.

In order to build a new classifier or modify a previous classifier on streaming unlabeled data, it is an intuitive method to periodically request a human expert to label a small amount of samples. To be used as a more practical method, the manual labeling of samples should be requested as little as possible.

We propose the method of forming an ensemble classifier on streaming unlabeled data. In order to build a new classifier for an ensemble, the proposed ensemble method selects unlabeled samples according to the change in distribution of streaming unlabeled data. If an ensemble “guesses” that some selected unlabeled samples belong to the same distribution, which is different from previous distributions, it requests a human expert to label them to build a new classifier.

2 Ensemble Approach for Streaming Unlabeled Data

An ensemble classifier consists of several models (classifiers). It predicts the class of a new sample by combining the predictions made by each classifier. Using a combining method to determine the final output, the weighted output of each classifier is combined.

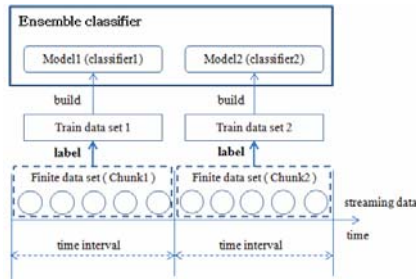


Fig. 1. Ensemble approach for classifying streaming unlabeled data

An ensemble classifier for classifying streaming data must determine when a new classifier should be built to maintain its classification accuracy. Also, to build a new classifier, an ensemble classifier must determine the samples which should be manually labeled. The typical ensemble approach partitions streaming data into a finite data set (chunk) using a predefined time interval to build a new classifier, as shown in Fig.1. The time interval is usually defined as the number of streaming samples [3,6,7,9-12]. In Fig.1, it is defined as 5. Such an approach can request a human expert to label all samples of a chunk periodically.

In order to select samples that could be labeled to build a new classifier for an ensemble, Zhu et al. [12] introduced three simple methods, and proposed the method called MV (Minimal Variance) using ensemble variance. All of these methods select the predefined number of samples from a chunk: (1) The random sampling (RS) method randomly selects those unlabeled samples; (2) The local uncertainty sampling (LU) method selects those unlabeled samples using an uncertainty measure of only the current chunk, without considering any other previous chunks; (3) The global uncertainty sampling (GU) method first labels a tiny set of samples from the current chunk and then builds a new classifier from them. Samples, that could be labeled, are selected from the unlabeled samples remaining in the current chunk using classifiers in an ensemble and the new classifier; (4) The minimal variance (MV) method only uses the ensemble variance as an uncertainty measure in the global uncertainty sampling (GU) method.

3 Suspicious Streaming Data

Suspicious streaming data is defined as “streaming unlabeled data that should be labeled to improve accuracy of the current classifier”. Suspicious data from streaming unlabeled data might be samples that belong to a distribution different from previous distributions, which are distributions of train data of each previous classifier in an ensemble. If a streaming sample belongs to a previous distribution, the sample may have a high probability of being correctly classified by the current classifier. We assume that a previous distribution is a normal distribution.

Suppose that streaming samples on d -dimensional space have belonged to k previous distributions until now. These streaming samples are separated into k data sets according to previous distributions, $S_1 = \{x_{11}, \dots, x_{1n_1}\}, S_2 = \{x_{21}, \dots, x_{2n_2}\}, \dots, S_k = \{x_{k1}, \dots, x_{kn_k}\}$. x_{ij} represents the j^{th} sample in the i^{th} data set. Each previous distribution N_i is summarized within the mean m_i and the standard deviation σ_i of the corresponding data set S_i . If a new streaming sample x' arrives, we can evaluate if the new sample x' belongs to one of the previous distributions using equation (1)

$$f(x', N_i) = \begin{cases} 1, & \text{if } \text{dist}(x', m_i) > \sigma_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where, $\text{dist}(x', m_i)$ represents the distance between the new sample x' and the mean m_i of N_i ($0 \leq i \leq k$).

If the new sample does not belong to any of the previous distributions, it becomes a suspicious sample.

Equation (2) is a distance function used in the proposed methodology. The distance function combines each normalized output of Euclidean, $\text{Euclidean}()$, and the cosine distance, $\text{Cosine}()$, functions according to each indicator, I_1, I_2 . Each indicator has a value of 0 or 1, and the output of this distance function is between 0 and 1. The Euclidean is used for numerical attributes, and the cosine distance is used for nominal (categorical) attributes. For example, if there is no nominal attribute in a streaming data, the indicator I_2 of the cosine distance function becomes 0.

$$\begin{aligned}
 dist(x_i, x_j) &= \frac{I_1}{I_1 + I_2} Eu(x_i^n, x_j^n) + \frac{I_2}{I_1 + I_2} Co(x_i^c, x_j^c) \\
 Eu(x_i^n, x_j^n) &= \frac{Euclidean(x_i^n, x_j^n)}{Euclidean_Max} \\
 Co(x_i^c, x_j^c) &= \frac{1}{d} \sum_{k=1}^d (1 - Cosine(x_i^c, x_j^c))
 \end{aligned} \tag{2}$$

x_i^n denotes a sample that consists of only numerical attributes of a sample x_i . x_i^c denotes a sample that consists of only nominal attributes of a sample x_i . In particular, each nominal attribute of x_i^c represents a frequency vector in which each element represents the count of its values. Suppose that “Color” is the nominal attribute and it has three values: {red, green, blue}. If “Color” value of x_i^c is red, then its frequency vector becomes (1,0,0). In a single sample, all frequency vectors become a unit vector. The *Euclidean_Max* represents the longest distance in training samples including the training samples for the initial classifier. The d denotes the number of nominal attributes.

When a mean vector of a previous distribution is calculated, the meaning of each nominal attribute becomes a frequency rate vector of the corresponding data set. For example, if a data set has 1000 samples, and the frequency vector of the “Color” attribute is (700,200,100), then the mean of the “Color” attribute becomes (0.7,0.2,0.1).

4 Ensemble Method Using Suspicious Streaming Samples

We propose an ensemble method of dynamically maintaining an ensemble classifier according to changes in distribution of streaming unlabeled data. The proposed ensemble consists of several classifiers, and the mean vectors and the standard deviations of train data for each classifier. The final classification output of an ensemble is obtained by weighting each classifier’s prediction. The proposed ensemble determines streaming unlabeled data that could be labeled from the suspicious streaming samples. Also, whenever an ensemble classifies a new sample, it evaluates weights of each classifier according to the relation of the new sample to previous distributions in the current ensemble; for further details see Ryu et al. [8]. This section describes how to build a new classifier for an ensemble using suspicious samples.

An initial ensemble consists of a single classifier that is built on a collected train data set in the offline mode. Constructing a new classifier for an ensemble including an initial classifier, an ensemble also summarizes the previous distribution (dotted circle) with the mean vector, m , and the standard deviation, σ , of its train data set (circles) as shown in Fig. 2. By the equation (1), a streaming sample (triangle) that is outside a previous distribution becomes a suspicious sample as depicted in the figure.

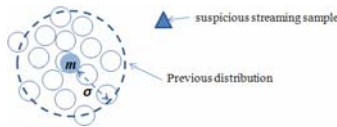


Fig. 2. Previous distribution and suspicious streaming sample in our ensemble method

The proposed ensemble forms a train set on suspicious streaming samples (triangle) that belong to the same distribution using the cluster concept as shown in Fig.3. Numbers inside the triangles denote the incoming order of stream samples. When the first suspicious sample occurs, it becomes the seed of a new cluster region. The radius of the new cluster region is settled as a predefined cluster radius, θ_r . The seed does not move until the number of samples within the 1st cluster region reaches the predefined minimum number of samples, θ_s . If the next suspicious sample (the 2nd suspicious sample) is located within this cluster region, it is assigned to the first cluster. If a new suspicious sample is located outside previous cluster regions as the 3rd suspicious sample in Fig.3, an ensemble generates another new cluster region.

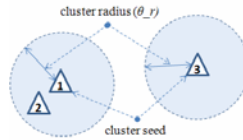


Fig. 3. Cluster region formed from suspicious samples

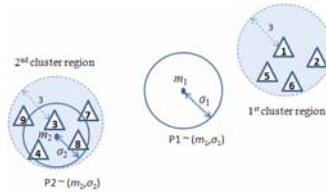


Fig. 4. Building a new classifier according to the change in distribution of streaming data

Within a cluster region, when the number of assigned samples reaches the predefined minimum number of samples, θ_s , an ensemble requests a human expert to label them. After they are labeled, an ensemble builds a new classifier on them.

The proposed ensemble method does not periodically build a new classifier for an ensemble. It builds a new classifier according to changes in distribution of streaming data as shown in Fig.4. Suppose that there is one previous distribution, $P1 \sim (m_1, \sigma_1)$, and nine suspicious streaming samples continuously occur. If θ_r and θ_s are 0.3 and 5, respectively, a new classifier is built on samples within the second cluster region after the 9th sample. Also, another previous distribution, $P2$ is defined with the mean vector, m_2 , and the standard deviation, σ_2 of these samples.

A value of θ_r is in the range of $[0, 1]$ as a result of the equation (2). If θ_s is defined as a large value, an ensemble has to wait for a new classifier until enough suspicious samples are collected as its train data.

5 Experiments

We evaluated the proposed ensemble method using synthetic streaming data and the intrusion detection data of KDD'99 Cup. Its results on the synthetic data sets are

compared with the results shown in Zhu et al. [12]. In the intrusion detection data, our method is compared with the ensemble method that requests correct classes of all samples gathered during a time interval. In all experiments, correct classes of samples were used whenever an ensemble builds a new classifier. The J48 decision tree (C4.5) from Weka (www.cs.waikato.ac.nz/ml/weka/) was used to build a new classifier of an ensemble in our experiments as well as Zhu et al. [12].

Synthetic data. We utilize the stream-data generation method used in Zhu et al. [12]. To generate streaming data, each parameter is settled with values used in Zhu et al. [12]. Each generated streaming data set has 50,000 samples with 10 dimensions. The concept drifting involves 5 attributes, and attribute weights change with a magnitude of 0.1 in every 1000 samples and weight adjustment inverts the direction with 20% of change. We generated 500 initial train samples with the first concept-drifting condition of streaming data before generating streaming data.

Intrusion detection data. This data set has been widely used in the stream data research [3]. It contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment (www.sigkdd.org). We used 10% subset of this data set, `kddcup.data_10_percent.zip` because Geo et al. [3] showed the changes in the data distribution in this subset. The subset has 494,020 samples with 41 attributes and 22 attack types. We transformed this subset into a data set with two classes (“normal” and “intrusion”) as in [3].

5.1 Results on Synthetic Streaming Data

In order to compare with results in Zhu et al. [12], we applied the proposed method to three types of synthetic streaming data: two-class, three-class, and four-class. Results are shown in Table 1. We also kept the most recent 10 classifiers in an ensemble. Each value in the table corresponds to average (\pm standard deviation) of (1) classification accuracies and (2) the total number of generated classifiers for an ensemble (classifier counts) of 10 experiments that used different data sets with same input parameters. Existing methods (random sampling (RS), local uncertainty (LU), global uncertainty (GU), and minimal variance (MV)) [12] periodically build a new classifier according to the predefined chunk size. Their classifier counts, therefore, have the same values. For example, when a chunk size is predefined as 250, the total number of generated classifiers becomes 200.

To build a new classifier, existing methods selected only samples of 10% from a chunk, and then used their correct classes. Therefore, we predefined the threshold, θ , as 25, 50, 75, 100, and 200. The value, 0.5, for θ , was decided through changes in measurements. At this value, both the classification accuracy and the classifier count have relatively small variances.

The proposed method shows higher classification accuracy and less classifier count than existing methods in many cases. In particular, when the number of samples that will be labeled (size of train data set) is 25, our ensemble produces average 26.9% higher accuracy and average 36.7% less classifier count. In the results of our ensemble, the larger the size of train data set is, the smaller the accuracy is. We will discuss this issue in Section 5.

Table 1. Results on synthetic data

| # classes | # samples that will be labeled for a new classifier | Proposed method ($\theta=0.5$) | | Active learning methods from streaming data [12] | | | | |
|---------------|-----------------------------------------------------|----------------------------------|-------------------------|--------------------------------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| | | Classifier count | % | Classifier count | RS (%) | LU (%) | GU (%) | MV (%) |
| Two classes | 25 | 164.1 (± 13.20) | 89.60 (± 1.61) | 200.0 (± 0.00) | 74.54 (± 2.89) | 73.70 (± 2.75) | 75.51 (± 3.35) | 81.08 (± 2.85) |
| | 50 | 79.1 (± 5.76) | 86.23 (± 2.08) | 100.0 (± 0.00) | 83.07 (± 2.42) | 82.56 (± 2.36) | 85.16 (± 2.95) | 86.79 (± 2.51) |
| | 75 | 54.4 (± 3.23) | 84.09 (± 2.64) | 66.0 (± 0.00) | 86.10 (± 2.75) | 85.69 (± 3.13) | 88.14 (± 3.04) | 88.67 (± 2.74) |
| | 100 | 43.6 (± 4.27) | 82.36 (± 2.03) | 50.0 (± 0.00) | 86.43 (± 3.52) | 86.11 (± 2.36) | 88.79 (± 3.47) | 89.09 (± 3.39) |
| | 200 | 25.4 (± 2.65) | 78.51 (± 2.41) | 25.0 (± 0.00) | 86.47 (± 5.01) | 85.94 (± 4.82) | 88.69 (± 4.27) | 88.88 (± 4.27) |
| Three classes | 25 | 162.6 (± 11.90) | 88.66 (± 0.70) | 200.0 (± 0.00) | 56.36 (± 2.71) | 55.46 (± 2.64) | 56.57 (± 2.71) | 64.41 (± 3.87) |
| | 50 | 78.1 (± 5.46) | 86.24 (± 1.53) | 100.0 (± 0.00) | 66.31 (± 4.30) | 66.11 (± 4.23) | 66.99 (± 4.98) | 71.59 (± 4.23) |
| | 75 | 58.7 (± 3.92) | 85.60 (± 1.58) | 66.0 (± 0.00) | 71.79 (± 2.86) | 70.49 (± 3.61) | 72.59 (± 3.77) | 74.39 (± 3.99) |
| | 100 | 44.8 (± 3.05) | 84.20 (± 1.26) | 50.0 (± 0.00) | 75.29 (± 2.88) | 74.23 (± 3.27) | 76.45 (± 3.72) | 78.22 (± 4.38) |
| | 200 | 25.5 (± 1.43) | 81.73 (± 0.91) | 25.0 (± 0.00) | 73.92 (± 6.11) | 72.97 (± 6.01) | 76.37 (± 4.64) | 75.98 (± 5.44) |
| Four classes | 25 | 163.2 (± 12.37) | 87.93 (± 1.21) | 200.0 (± 0.00) | 42.24 (± 2.01) | 41.38 (± 1.80) | 41.34 (± 2.09) | 46.54 (± 2.04) |
| | 50 | 81.1 (± 5.18) | 85.79 (± 0.89) | 100.0 (± 0.00) | 50.47 (± 2.34) | 49.81 (± 2.47) | 49.91 (± 2.45) | 53.95 (± 3.04) |
| | 75 | 54.4 (± 4.05) | 85.18 (± 2.08) | 66.0 (± 0.00) | 58.11 (± 3.65) | 56.75 (± 3.26) | 56.48 (± 3.09) | 59.86 (± 3.88) |
| | 100 | 44.4 (± 4.71) | 84.87 (± 1.71) | 50.0 (± 0.00) | 63.08 (± 3.67) | 62.72 (± 4.07) | 61.92 (± 3.56) | 64.04 (± 4.11) |
| | 200 | 26.0 (± 2.23) | 80.77 (± 1.60) | 25.0 (± 0.00) | 71.87 (± 4.47) | 70.67 (± 4.98) | 70.98 (± 5.09) | 72.13 (± 5.33) |

The proposed method also produced 0.46 of standard deviation on accuracies (84.15%, 85.28%, and 84.90%) that are averaged per synthetic streaming data type. This standard deviation value is smaller than that of existing methods: (SV, 10.70), (LU, 10.86), (GU, 11.90), (MV, 11.27). This shows that our method is much more robust than existing methods in a multi-class problem.

5.2 Results on Intrusion Detection Data

In Table 2, we report performances of ensemble methods including existing methods (simple voting (SV) [3], weighted ensemble (WE) [11]) on the intrusion detection data. When building a new classifier, existing methods use the correct classes of “all” samples within a chunk. To determine the final output of an ensemble, the SV selects the most frequent class from a set of classes that is predicted by each classifier. The WE combines predictions of each classifier per a class as the weighted average, and then the class with the greatest average value is selected. In the WE, weights of each classifier in an ensemble are evaluated as their classification accuracy on the most recent chunk.

For two parameter values: (1) the number of samples that will be labeled, and (2) the maximum number of classifiers in ensemble, we selected the values used in the synthetic streaming data. In an ensemble, when there are over 10 classifiers, our method and the SV keep 10 classifiers built at the most recent time. The WE deletes one with the lowest weight after evaluating weights of each classifier.

Table 2. Results on intrusion detection data

| # samples that will be labeled for a new classifier | Proposed method ($\theta_r = 0.3$) | | Ensemble methods | | |
|-----------------------------------------------------|--------------------------------------|----------------------|----------------------------|----------------------|----------------------|
| | Generated classifier count | % | Generated classifier count | SV [3] (%) | WE [11] (%) |
| 25 | 320 | 97.04 | 19,414 | 98.46 | 99.31 |
| 50 | 104 | 96.61 | 9,707 | 97.82 | 98.91 |
| 75 | 10 | 97.38 | 6,471 | 97.18 | 98.59 |
| 100 | 9 | 97.43 | 4,853 | 96.54 | 98.31 |
| 200 | 11 | 92.98 | 2,426 | 95.09 | 97.30 |
| Average | 90.80 (120.24) | 96.29 (± 1.87) | 8,574.20 (± 5913.63) | 97.02 (± 1.29) | 98.48 (± 0.75) |

In Table 2, a value of θ_r , is selected as 0.3. At this value, our method shows a balanced trade-off between classification accuracy and labeling efficiency. With average 98.9% (8483.4) lesser new classifiers for the intrusion detection data, the proposed method produced similar accuracy to existing methods. If the chunk size is predefined as larger value, the existing methods can produce smaller value of the classifier count. However, the more number of samples should be labeled.

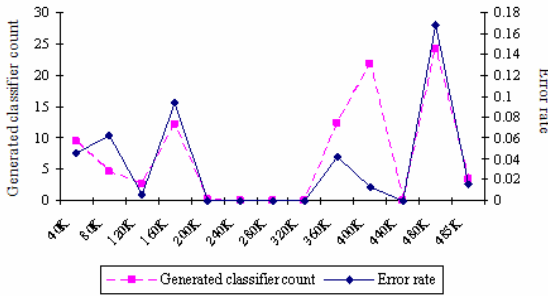


Fig. 5. Classifier count and error rate on streaming data

The first 8,653 samples (i.e 1.7%) of the intrusion detection data (kddcup.data_10_percent.zip) are used to build the initial classifier for an ensemble. This initial train data set includes about 10% of samples with “intrusion” class. Figure 5 shows the distribution in time of an average classifier count and an average error rate per 40K streaming data. The error rate represents the proportion of the number of misclassified samples to 40K samples. Both values of the average classifier count and the average error rate increase in sections where the ratio of class is changed such as 0K~40K, 120K~160K, 320K~360K, and 440K~480K. In the section between 40K and 80K, the average error rate increases, while the average classifier count reduces

because an ensemble delays building a new classifier until the number of suspicious samples reaches the threshold, θ_s . In the section between 160K and 320K, any requesting correct classes of samples did not occur because most samples within this section might belong to previous distributions in the current ensemble.

6 Discussion

Table 1 is the experimental results on synthetic streaming data. When the number of samples that will be labeled has a large value, the classification accuracy of the proposed method decreases, while that of the existing methods increase. Our method builds a new classifier according to changes in a distribution of streaming data, not at a time interval. To compare with results in Zhu et al. [12], we deleted the oldest classifier in the current ensemble when the number of its classifiers is over the predefined maximum number. Accordingly, the oldest classifier is removed from the current ensemble only if a new classifier is added to the current ensemble. In our method, if the size of a training data set is large, the current ensemble may contain previous distributions, where the change in a class distribution occurs, and the corresponding classifiers. They may negatively affect classification accuracy of the current ensemble. The class distribution means the areas with high density of each class inside a previous distribution region. Also, that delete mechanism may remove a previous distribution with an unchanged class distribution and the corresponding classifier.

7 Conclusions

We proposed an ensemble method of dynamically forming an ensemble on streaming unlabeled data. To build a new classifier, the proposed ensemble method selects samples that will be labeled according to changes in streaming data distribution. Those unlabeled samples are defined as suspicious streaming samples that do not belong to any previous distribution in the current ensemble.

Our method constructed an ensemble more efficiently than existing ensemble methods that periodically build a new classifier according to a time interval. From synthetic streaming data sets, our method produced average 14.1% higher classification accuracy than existing ensemble methods [12], and the number of new classifiers for an ensemble reduced by average 12.6%. In particular, we showed that our method is much more robust than existing methods in a multi-class problem. Also, for the real-world problem (intrusion detection data set), our method produced similar average 96.29% accuracy to existing methods [3,11] using correct labels of only 0.007%(3410) samples of 485,377 streaming unlabeled data. While, existing methods [3,11] use correct labels of all streaming unlabeled data. We believe that in order to apply an ensemble method to real-world streaming data, an ensemble should demand correct classes of samples as little as possible.

In this paper, our method deleted the oldest classifier within the current ensemble. As mentioned in the discussion section, the delete mechanism considering only time feature might negatively affect the performance of an ensemble. Therefore, we are planning to extend the proposed method to delete a useless classifier in the current ensemble using other information, such as density, as well as time information.

Acknowledgments. This research has been partially funded by National Science Foundation (NSF) under grant #0637563 and Kentucky Science and Technology Corp. (KSTC) under grant #KSTC-144-401-07-018.

References

1. Aggarwal, C.C.: *Data Streams: Models and Algorithms*. Springer, Heidelberg (2007)
2. Cozman, F.G., Cohen, I.: Unlabeled Data Can Degrade Classification Performance of Generative Classifiers. In: 15th International Florida Artificial Intelligence Society Conf., USA, pp. 327–331 (2002)
3. Gao, J., Fan, W., Han, J.: On Appropriate Assumptions to Mine Stream: Analysis and Practice. In: Seventh IEEE International Conf. on Data Mining, USA, pp. 143–152 (2007)
4. Kolter, J.Z., Maloof, M.A.: Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts. *Journal of Machine Learning Research* 8, 2755–2790, Microtome (2007)
5. Kuncheva, L.I., Whitaker, C.J., Narasimhamurthy, A.: A case-study on naïve labeling for the nearest mean and the linear discriminate classifiers. In: *Pattern Recognition*, vol. 41, pp. 3010–3020. Elsevier, Amsterdam (2008)
6. Katakis, I., Grigorios, T., Vlahavas, I.: An Ensemble of Classifiers for coping with Recurring Contexts in Data Streams. In: 18th European Conf. on Artificial Intelligence, Greece, pp. 763–764 (2008)
7. Masud, M.M., Gao, J., Khan, L., Han, J., Thuraisingham, B.: A Multi-partition Multi-chunk Ensemble Technique to Classify Concept-Drifting Data Streams. In: Theeramunkong, T., Kijssirikul, B., Cercone, N., Ho, T.-B. (eds.) *PAKDD 2009. LNCS (LNAI)*, vol. 5476, pp. 363–375. Springer, Heidelberg (2009)
8. Ryu, J.W., Kantardzic, M., Walgampaya, C.: Ensemble Classifier based on Misclassified Streaming Data. In: *Proceeding of the 10th IASTED International Conf. on AIA*, Austria, pp. 347–354 (2010)
9. Scholz, M., Klinkenberg, R.: An Ensemble Classifier for Drifting Concepts. In: *Proceeding of the Second International Workshop on Knowledge Discovery in Data Streams*, Portugal, pp. 53–64 (2005)
10. Street, W.N., Kim, Y.: A Streaming Ensemble Algorithm (SEA) for Large-Scale Classification. In: *Proceeding of the 7th ACM International Conf. on Knowledge Discovery and Data Mining*, New York, pp. 377–382 (2001)
11. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining Concept-Drifting Data Streams using Ensemble Classifiers. In: *Proceedings of ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining, USA*, pp. 226–235 (2003)
12. Zhu, X., Zhang, P., Lin, X., Shi, Y.: Active Learning from Data Streams. In: *Seventh IEEE International Conf. on Data Mining, USA*, pp. 757–762 (2007)

Random Projections for SVM Ensembles

Jesús Maudes, Juan J. Rodríguez, César García-Osorio, and Carlos Pardo

University of Burgos, Spain

{jmaudes, jjrodriguez, cgosorio, cpardo}@ubu.es

Abstract. Data projections have been used extensively to reduce input space dimensionality. Such reduction is useful to get faster results, and sometimes can help to discard unnecessary or noisy input dimensions. Random Projections (RP) can be computed faster than other methods as for example Principal Component Analysis (PCA). This paper presents an experimental study over 62 UCI datasets of three types of RPs taking into account the size of the projected space and using linear SVMs as base classifiers. We also combined random projections with sparse matrix strategy used by Rotation Forests, which is a method based in projections too. Results shows that Random Projections use to be better than using PCA for SVMs ensembles.

Keywords: Ensembles, Random Projections, Rotation Forests, Diversity, Kappa-Error Relative Movement diagrams.

1 Introduction

Projections techniques are broadly used to reduce input dimensionality in classification problems. Projection methods are designed to preserve in some way data original structure in the projected space, so projected data can be used to speed up classifiers training and sometimes help to avoid noise and over-fitting. PCA is probably the most popular projection method. It is used to reduce data dimensionality capturing a percentage of the variance in the original data. The main drawback of PCA is also its computational complexity. Random Projections (RP) [1], [2] have a lower computational cost. Some RPs can maintain pairwise distances in the projected space within an arbitrary small factor.

Support Vector Machines (SVM) [3] are very accurate and stable classifiers. Small changes in the training dataset does not make very different SVMs. Therefore, it is difficult to get an ensemble of SVMs that performs better than a single SVM using state of art ensemble methods. One question to answer in this paper is if randomness inherent to RPs can be considered as a source of diversity that aims at getting accurate SVM ensembles. So, we are not interested in using projection for reducing input dimensionality but to increase SVM ensembles performance.

Rotation Forests [4] is an ensemble method for decision trees. It uses PCA to project different groups of attributes in each base classifier. In [5] is shown that PCA is better than RP for such ensemble method of decision trees. However,

the essential ingredient of Rotation Forests is to project using those groups of attributes, which also makes base classifiers in the ensemble are trained different each other. That difference in training process does not produce very diverse base classifiers, but it seems to keep their individual accuracy. In [5] diversity is analyzed for RPs with and without splitting into groups of attributes. Projecting without splitting the input space turn into more diverse classifiers but also less accurate. Hence, *the additional diversity obtained through the “full” random projection is not useful for the ensemble*. That work was made for decision trees which are very sensitive to small changes. In our work we also test the effect of split projections using SVM as base classifiers.

The rest of the paper is organized as follows: Random Projections considered in this work are described in Section 2. An experimental study is presented in Section 3. Analysis of diversity is in Section 4. Finally, conclusions are summarized in Section 5.

2 Random Projections

For projecting data a transformation matrix is needed. When an instance x is projected the vector containing its values is multiplied by this matrix obtaining a new vector (i.e. x projection). In random projections the matrix entries are random numbers. In this work three types of random projections have been used.

1. Each entry in transformation matrix comes from a Gaussian random generator. This RP is denoted as *Gaussian* in this work.
2. The entries values are $\sqrt{3} \times x$, where x is a random number taking the following values: -1 with probability $1/6$, 0 with probability $2/3$ and $+1$ with probability $1/6$. This RP is denoted as *Sparse* in this work.
3. The entries are -1 with probability $1/2$ and $+1$ with probability $1/2$. This RP is denoted as *Binary* in this work.

The two latter are described in [1]. They are based on Johnson and Lindenstrauss theorem [6]. This theorem states that *given $\epsilon > 0$, an integer n and k a positive integer such that $k \geq k_0 = O(\epsilon^{-2} \log(n))$. For every set P of n points in \mathbb{R}^d there exists $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$ such that for all $u, v \in P$*

$$(1 - \epsilon)\|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \epsilon)\|u - v\|^2 \quad (1)$$

Hence, these two methods are fast to compute and aim at preserving pairwise euclidean distances in the projected space.

It is expected that Sparse projection can contribute to diversity because the amount of zeros. The zeros would reject some original dimensions in the computation of some of the new dimensions. There are successful ensemble methods that also trains their base classifiers by excluding some existing features. In the Random Subspaces method [7] each base classifier only takes into account a subset of the attributes from the original space. The size of this subset of attributes is specified as a percentage.

3 Experiments

Experimental validation has been made using WEKA [8] for the ensembles and projections. In order to limit the analysis scope to a manageable number of combinations, linear kernel was the only kernel considered for the SVMs in the experiment. LIBLINEAR [9] was used because it provides a fast linear kernel SVM implementation. Default parameters were used in all methods where not indicated.

The 62 datasets from UCI repository [10] used in the experiments are shown in Table 1. Nominal attributes are computed using Nominal to Binary transformation in all methods.

Table 1. Summary of the data sets used in the experiments

| Dataset | #N | #D | #I | #E | #C | Dataset | #N | #D | #I | #E | #C |
|---------------|----|-----|-----|-------|----|-----------------|-----|----|-----|-------|----|
| abalone | 7 | 1 | 10 | 4177 | 28 | lymphography | 3 | 15 | 38 | 148 | 4 |
| anneal | 6 | 32 | 90 | 898 | 6 | mushroom | 0 | 22 | 121 | 8124 | 2 |
| audiology | 0 | 69 | 93 | 226 | 24 | nursery | 0 | 8 | 26 | 12960 | 5 |
| autos | 15 | 10 | 71 | 205 | 6 | optdigits | 64 | 0 | 64 | 5620 | 10 |
| balance-scale | 4 | 0 | 4 | 625 | 3 | page | 10 | 0 | 10 | 5473 | 5 |
| breast-w | 9 | 0 | 9 | 699 | 2 | pendigits | 16 | 0 | 16 | 10992 | 10 |
| breast-y | 0 | 9 | 48 | 286 | 2 | phoneme | 5 | 0 | 5 | 5404 | 2 |
| bupa | 6 | 0 | 6 | 345 | 2 | pima | 8 | 0 | 8 | 768 | 2 |
| car | 0 | 6 | 21 | 1728 | 4 | primary | 0 | 17 | 23 | 339 | 22 |
| credit-a | 6 | 9 | 43 | 690 | 2 | promoters | 0 | 57 | 228 | 106 | 2 |
| credit-g | 7 | 13 | 61 | 1000 | 2 | ringnorm | 20 | 0 | 20 | 300 | 2 |
| crx | 6 | 9 | 42 | 690 | 2 | sat | 36 | 0 | 36 | 6435 | 6 |
| dna | 0 | 180 | 180 | 3186 | 3 | segment | 19 | 0 | 19 | 2310 | 7 |
| ecoli | 7 | 0 | 7 | 336 | 8 | shuttle | 9 | 0 | 9 | 58000 | 7 |
| glass | 9 | 0 | 9 | 214 | 6 | sick | 7 | 22 | 33 | 3772 | 2 |
| heart-c | 6 | 7 | 22 | 303 | 2 | sonar | 60 | 0 | 60 | 208 | 2 |
| heart-h | 6 | 7 | 22 | 294 | 2 | soybean | 0 | 35 | 84 | 683 | 19 |
| heart-s | 5 | 8 | 25 | 123 | 2 | soybean-small | 0 | 35 | 84 | 47 | 4 |
| heart-statlog | 13 | 0 | 13 | 270 | 2 | splice | 0 | 60 | 287 | 3190 | 3 |
| heart-v | 5 | 8 | 25 | 200 | 2 | threenorm | 20 | 0 | 20 | 300 | 2 |
| hepatitis | 6 | 13 | 19 | 155 | 2 | tic-tac-toe | 0 | 9 | 27 | 958 | 2 |
| horse-colic | 7 | 15 | 60 | 368 | 2 | twonorm | 20 | 0 | 20 | 300 | 2 |
| hypo | 7 | 18 | 25 | 3163 | 2 | vehicle | 18 | 0 | 18 | 846 | 4 |
| ionosphere | 34 | 0 | 34 | 351 | 2 | vot1 | 0 | 15 | 45 | 435 | 2 |
| iris | 4 | 0 | 4 | 150 | 3 | voting | 0 | 16 | 16 | 435 | 2 |
| krk | 6 | 0 | 6 | 28056 | 18 | vowel-context | 10 | 2 | 26 | 990 | 11 |
| kr-vs-kp | 0 | 36 | 40 | 3196 | 2 | vowel-nocontext | 10 | 0 | 10 | 990 | 11 |
| labor | 8 | 8 | 26 | 57 | 2 | waveform | 40 | 0 | 40 | 5000 | 3 |
| led-24 | 0 | 24 | 24 | 5000 | 10 | yeast | 8 | 0 | 8 | 1484 | 10 |
| letter | 16 | 0 | 16 | 20000 | 26 | zip | 256 | 0 | 256 | 9298 | 10 |
| lrd | 93 | 0 | 93 | 531 | 10 | zoo | 1 | 15 | 16 | 101 | 7 |

#N: Numeric features, #D: Discrete features, #E: Examples, #I:Inputs, #C: Classes

Random Projections were used in the following ensembles:

1. An ensemble of SVMs trained with projected data. Three sizes of projected space dimension have been tested (i.e. 75%, 100% and 125% of attributes). The ensemble computes its prediction as the straight average of the probabilities predicted by its members. These configurations are denoted as *RP-Ensemble n%*, where n is the percentage indicating the dimension of the projected space. Hence 125% configuration makes input dimensionality grow.
2. A Rotation Forests [4] variant that replaces PCA projection by RPs and the Decision Trees by SVMs. Rotation Forests divides input space into different partitions for each base classifier. In this experiment the size of all partitions has been set to 5. For each partition an RP is computed. The sizes of the projections tested has been set again to 75%, 100% and 125% of the 5 attributes. Again we want to test projections that augment the original problem dimension. These configurations has been denoted as *Rot-RP n%*, where n is the percentage indicating the dimension of the projected partitions. For Rot-RP all nominal attributes have been previously converted into binary to increase the number of partitions. Column #I in Table 1 shows the resulting dimensionality after such conversion.

These 6 configurations have been tested using the 3 random projections described in previous section (i.e. Gaussian, Sparse, and Binary, denoted as G, S and B respectively), resulting into 18 RP based ensembles.

These RP based ensembles have been tested against the base method on its own, (i.e. LIBLINEAR SVM) and the following state of art ensemble methods:

- Bagging [11] of SVM.
- Boosting of SVM. AdaBoost [12] and MultiBoost [13] versions were considered. Resampling version of both methods was used (i.e. training instances for each base classifier are obtained according its weights distribution). The number of subcommittees in MultiBoost was set to 10.
- Random Subspaces [7] using 50% and 75% of original features (i.e. each nominal feature is considered as one feature).
- Rotation Forests [4] replacing Decision Trees by SVM. Two configurations was tested changing the parameter that controls the proportion of variance retained by PCA projection, which was set to 75% and 100%. These configurations are denoted as *Rot-PCA* in tables. In order to compare Rotation PCAs to Rotation-RPs the size of attribute partitions was also set to 5, and nominal to binary conversion is applied beforehand as well.

All ensemble configurations in the experiment use 50 base SVM classifiers. The results were obtained using 5×2 stratified cross validations.

Table 2 shows all the methods ordered by their average ranks [14], “*average ranks by themselves provide a fair comparison of the algorithms*”. Average ranks are computed sorting all the methods by their accuracy for each dataset. Then the average position of each method through all datasets is assigned as average rank for that method. The six best ranked methods in the table are Rot-RPs

Table 2. Average rank of the considered methods. *Avg Rank* column represents the average position taken by the method through all datasets. The methods are ordered by this column. *Pos* points the method position according such order.

| Pos | Method | Avg Rank | Pos | Method | Avg Rank |
|-----|----------------------|----------|------|----------------------|----------|
| 1 | Rot-RP 125%, B | 9.21 | 14 | RP-Ensemble 125%, G | 13.73 |
| 2 | Rot-RP 100%, G | 9.23 | 15.5 | Rot-RP 75%, G | 14.69 |
| 3 | Rot-RP 125%, S | 9.52 | 15.5 | Rot-RP 75%, B | 14.69 |
| 4 | Rot-RP 125%, G | 9.85 | 17 | Rot-PCA 75% | 15.01 |
| 5 | Rot-RP 100%, S | 10.47 | 18 | RP-Ensemble 100%, G | 15.10 |
| 6 | Rot-RP 100%, B | 10.88 | 19 | RP-Ensemble 125%, B | 15.19 |
| 7 | Rot-PCA 100% | 10.93 | 20 | AdaBoost | 15.69 |
| 8 | Bagging | 11.28 | 21 | Rotation-RP 75%, S | 16.02 |
| 9.5 | RP-Ensemble 125%, S | 12.34 | 22 | RP-Ensemble 75%, S | 16.14 |
| 9.5 | SVM | 12.34 | 23 | RP-Ensemble 100%, B | 16.31 |
| 11 | Random Subspaces 75% | 12.47 | 24 | Random Subspaces 50% | 16.90 |
| 12 | MultiBoost | 12.79 | 25 | RP Ensemble 75%, G | 17.64 |
| 13 | RP-Ensemble 100%, S | 13.28 | 26 | RP Ensemble 75%, B | 19.01 |

configurations followed by Rot-PCA 100% and Bagging. Moreover, configurations projecting 125% of attributes from these six configurations can get better results than some configurations projecting 100%. There is a tie between SVM and RP-Ensemble 125% S, and the rest of methods are placed behind SVM. Projections reducing the input space get the bottom places.

Table 3 shows wins, ties and losses of best Rot-RPs against SVM and the rest of ensembles ranked better or equal than SVM. Every Rot-RP configuration wins the RP-Ensemble configuration in the last row. Regarding the rest of rows (i.e. SVM, Bagging, Rot-PCA 100%) Sparse and Binary projections for Rot-RP 100% show the worse results. Differences of these Rot-RP configurations with the other three methods range from one to two victories. However, there are bigger differences when using Rot-RP 100% with Gaussian projections or with Rot-RP 125% with any type of random projection. For these four Rot-RP configurations there are always more wins than losses when compared with the other methods.

According to [14] using a sign test one method is significantly better than other, with a confidence level of 0.05, if the number of wins plus half the ties is at least $N/2 + 1.96\sqrt{N}/2$. For $N = 62$ datasets, the number of necessary wins is 39. Table 3 shows:

1. RP-Ensemble 125% S loses significantly against all configurations of Rot-RP 100% and 125%. This result agrees with [4] where experiments states that splitting input space is essential.
2. Stand alone SVM significantly loses against the 125% Rot-RP configurations.
3. Rot-PCA 100% is also very near to lose significantly against Rot-RP 125% S. This result is apparently in contradiction with [4] where ingredients of Rotation Forests method are analyzed. In that work PCA is better than RP for decision trees as base classifiers.

Table 3. Win-Ties-Losses of Rotation-RPs variants against the rest of best ranked methods. Boldface is used to point the winner. The symbol \bullet means a significantly win.

| | Rot-RP 100% | | | Rot-RP 125% | | |
|---------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| | Sparse | Binary | Gaussian | Sparse | Binary | Gaussian |
| Rot-PCA 100% | 28-5- 29 | 29-3- 30 | 30 -7-25 | 35 -7-20 | 32 -5-25 | 31 -6-25 |
| Bagging | 30 -3-29 | 28-4- 30 | 32 -4-26 | 31 -6-25 | 34 -5-23 | 30 -5-27 |
| SVM | 30 -4-28 | 29 -5-28 | 35 -4-23 | \bullet 37 -6-19 | \bullet 35 -8-19 | \bullet 37 -5-20 |
| RP-Ensemble 125%, S | \bullet 44 -2-16 | \bullet 37 -5-20 | \bullet 43 -4-15 | \bullet 39 -6-17 | \bullet 43 -7-12 | \bullet 40 -4-18 |

4 Kappa-Error Analysis

Ensembles success requires both accuracy and diversity from its base members. Kappa-error diagrams [15] are used to show how much accuracy and diversity is in base classifiers of an ensemble for a given dataset. In these diagrams a cloud is plotted. Each point of the cloud represents a pair of base classifiers. The coordinate x of a point is a measure of diversity between these base classifiers (i.e. κ), and the y coordinate is the average error from both classifiers.

κ takes their values from -1 to $+1$. $+1$ means both classifiers agree always, 0 means that there is the same level of agreement as if outputs were random, and -1 means higher disagreements, which are not usual. So a cloud of points at bottom left corner means that base classifiers of an ensemble are accurate and diverse.

Each kappa-error diagram can only contain a few clouds. Usually two clouds are plotted in each diagram representing the behavior of two ensembles for an only dataset. To show the behavior of 62 datasets kappa-error relative movement diagrams [16] are more suitable. Figure 1 shows these diagrams for our experiment. In these diagrams each arrow represents a dataset. Each diagram compares overall accuracy and diversity between two ensembles.

Computation of kappa-error relative movement diagrams is as follows:

1. For each dataset and ensemble in the study centers of clouds from kappa error diagrams are computed. For diagrams in this work, clouds come from computing 5×2 cross validation.
2. An arrow is drawn for each dataset connecting the centers of each pair of ensembles to compare.
3. Finally, all arrows are taken to the origin of coordinates.

Therefore, if arrows that come from an ensemble A clouds to an ensemble B clouds are taking bottom left direction, it means that base classifiers in B are more accurate and diverse than in A. In Figure 1 all diagrams compare one of the most successful ensembles in the study against the winner of the average rank (i.e. Rot-RP 125% B). Arrows are prone to point bottom right corner in all diagrams, so it means that base classifiers in the winner ensemble are less diverse than in its competitors, but slightly more accurate.

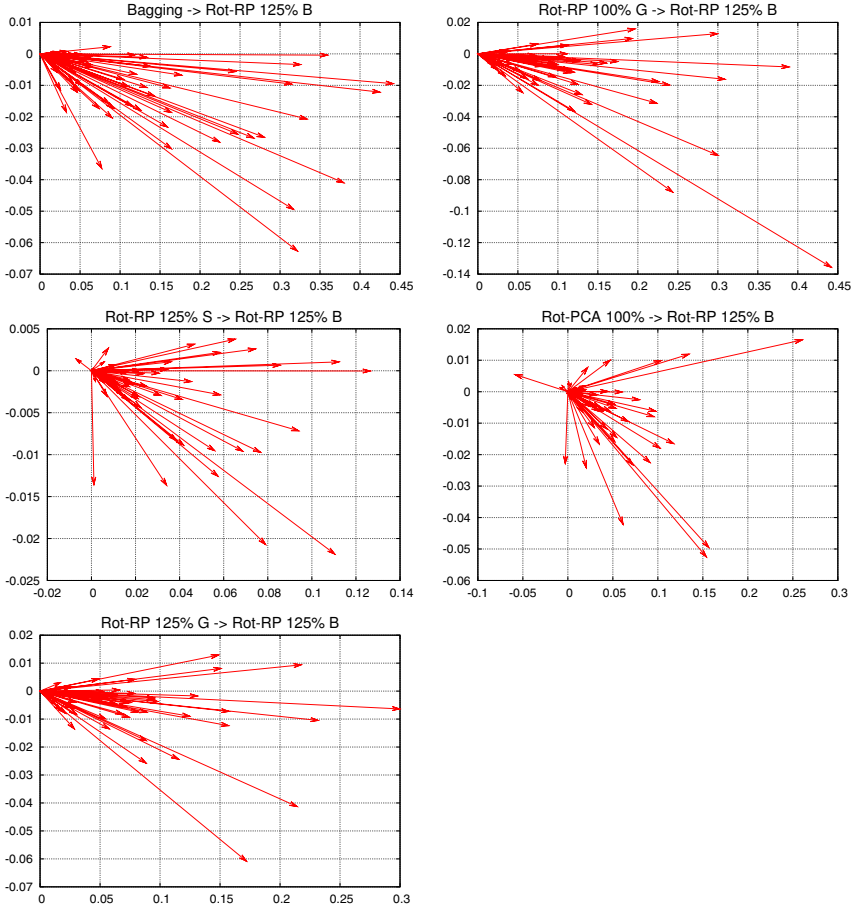


Fig. 1. Kappa error relative movement diagrams

5 Conclusions

In this work RPs are used to build ensembles of SVMs. Three types of RPs were tested (i.e. Gaussian, Sparse, Binary) against traditional PCA technique and other state of art ensemble methods. Projections were applied in two ways: (i) transforming the whole instance with the same projection and (ii) splitting the instance into groups of attributes and projecting each group with a different projection. The latter strategy is taken from the Rotation Forest method.

In the experiments, configurations that reduce input dimensionality perform worse than the ones that keep or augment it. Projecting the instances without splitting also leads to poor results. Rot-RFs 125% and 100% are the best ranked configurations. However, a further analysis has uncovered that 125%

configurations seems to perform slightly better than 100% configurations. Kappa-error relative movement diagrams show that best configuration does not lead to more diverse base classifiers, but more accurate.

Acknowledgements

This work was supported by the Project TIN2008-03151 of the Spanish Ministry of Education and Science.

We wish to thank the developers of WEKA and LIBLINEAR. We also express our gratitude to the donors of the different datasets and the maintainers of the UCI Repository.

References

1. Achlioptas, D.: Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *J. Comput. Syst. Sci.* 66(4), 671–687 (2003)
2. Fradkin, D., Madigan, D.: Experiments with random projections for machine learning. In: *KDD 2003: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 517–522. ACM, New York (2003)
3. Vapnik, V.N.: *The Nature of Statistical Learning Theory* (Information Science and Statistics). Springer, Heidelberg (1999)
4. Rodríguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(10), 1619–1630 (2006)
5. Kuncheva, L.I., Rodríguez, J.J.: An experimental study on rotation forest ensembles. In: Haindl, M., Kittler, J., Roli, F. (eds.) *MCS 2007*. LNCS, vol. 4472, pp. 459–468. Springer, Heidelberg (2007)
6. Johnson, W., Lindenstrauss, J.: Extensions of Lipschitz maps into a Hilbert space. In: *Conference in Modern Analysis and Probability*. Yale University, New Haven and London (1982); Vol. 26 of *Contemporary Mathematics*, AMS, 189–206 (1984)
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. Witten, I., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005), <http://www.cs.waikato.ac.nz/ml/weka>
9. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. *Journal of Machine Learning Research* 9, 1871–1874 (2008)
10. Asuncion, A., Newman, D.: UCI machine learning repository (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
11. Breiman, L.: Bagging predictors. *Machine Learning* 24(2), 123–140 (1996)
12. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1), 119–139 (1997)
13. Webb, G.I.: Multiboosting: A technique for combining boosting and wagging. *Machine Learning* 40(2) (2000)

14. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
15. Margineantu, D.D., Dietterich, T.G.: Pruning adaptive boosting. In: *Proc. 14th International Conference on Machine Learning*, pp. 211–218. Morgan Kaufmann, San Francisco (1997)
16. Maudes, J., Rodríguez, J.J., García-Osorio, C.: Disturbing neighbors diversity for decision forests. In: Okun, O., Valentini, G. (eds.) *Workshop on Supervised and Unsupervised Ensemble Methods and their Applications, SUEMA 2008*, pp. 67–71 (2008)

Rotation Forest on Microarray Domain: PCA versus ICA

Carlos J. Alonso-González, Q. Isaac Moro-Sancho, Iván Ramos-Muñoz,
and M. Aránzazu Simón-Hurtado

Intelligent Systems Group (GSI), Department of Computer Science,
E.T.S.I Informática, University of Valladolid, Valladolid, Spain
{calonso, isaac, ivan, arancha}@infor.uva.es

Abstract. Rotation Forest (RF) is an ensemble method that has shown effectiveness on microarray data set classification problems. RF works by generating sparse rotation matrixes of the input space, a method that creates accurate and diverse base classifiers. In its original formulation, elemental rotations were obtained by Principal Component Analysis (PCA). However, for microarray data sets, Independent Component Analysis (ICA) may be a better option. In this paper, an experimental study on ten microarray data sets has been performed. The study confirms that, except for a small number of attributes, Rotation Forest outperforms Bagging and Boosting on this domain. However, RF with ICA does not generally improve on RF with PCA.

Keywords: Rotation Forest, FastICA, microarray gene expression, attribute selection.

1 Introduction

The application of machine learning methods to DNA microarray gene expression data allows for systematic and high throughput analysis procedures, compared to histological and other methods [24]. Microarray classification is used to discover discriminating genes that allow identifying tissue types and it has been intensively applied to cancer diagnosis. However, microarray data classification is a challenging issue because of its high dimensionality and the small sample sizes. Typical values are around 10.000 gene expressions and a hundred or less tissue samples. To tackle this problem, the need to reduce dimensionality was soon recognized and the use of feature selection techniques has become customary [22]. Since the pioneering work of [7], a plethora of methods have been proposed combining different feature selection methods with different classification techniques.

Recently, interest has aroused on the application of ensemble methods to microarray classification problems. An up to date review is provided in [19]. Ensemble methods combine the output of several individual classifiers to obtain predictions that are usually more precise and robust than a standalone classifier

in several domains [12]. Among the newly proposed ensemble methods, Rotation Forest [21,11] seems to be an effective method for microarray classification, particularly when few genes are retained [23].

Rotation Forest generates precise ensembles because it finds a good tradeoff between accuracy and diversity of the classifiers it combines. The basic idea of Rotation Forest consists of projecting the input space on a random partition of the attribute set, generating axes rotations on each partition that are aggregated to create a new input space of the same dimension as the original, but where axes have been rotated. Originally, Principal Components Analysis, PCA, [9], keeping all components, was used to generate the axes rotation. However, recent works suggest that, for microarray classification problems, Independent Component Analysis (ICA) [10], may be a better option. To the best of our knowledge, there are two works [19,17] that found that Rotation Forest using ICA to transform the axes is preferable to Rotation Forest using PCA. However, these experimental studies are limited because they focus on one or two microarray data sets.

In this work, we extend this comparison to ten data sets to experimentally test if Rotation Forest with ICA is to be preferred to Rotation Forest with PCA on microarray classification problems. We focus on classification when a small number of genes is selected, because there is evidence that a small number of genes are sufficient [7,15,27].

The rest of the paper is organized as follows. Section 2 briefly describes Rotation Forest, provides some insight on the motivation of ICA for microarray gene expression problems and discusses the attribute selection methods employed. Section 3 describes the data sets and the experimental setting, while section 4 presents and discusses the experimental results. Section 5 summarizes the conclusions.

2 Rotation Forest

Rotation Forest is an ensemble method introduced in [21] that builds homogeneous ensembles by feature axes rotation. Hence, base classifiers must be sensitive to axes rotations, which makes decision trees a common choice as base classifiers.

The key idea of Rotation Forest is building accurate base classifiers using all features to construct each classifier, introducing diversity by generating different axes rotations. If instances of the learning problem are described by n features, Rotation Forest randomly selects K disjoint subsets of M features. In the original formulation, PCA is used to create an axes rotation on each K subset. No feature selection is made, because all components are preserved. A rotation matrix R_i is built aggregating the rotation found for each K subset and rearranging it according to the original order of the features in the instance space. The original training set is processed by matrix R_i to generate the training set T_i , which is used to induce the base classifier C_i . Due to the fact that probability of having different rotation matrix R_i for a large number of classifier may be small, two additional heuristics are employed to favor base classifiers diversity. Instead of performing PCA projecting the whole training data on each K subset of features, a nonempty subset of classes is randomly selected for each K subset of features, and PCA is performed on K bootstrap samples of 75% of the training data.

As indicated in [11], splitting the feature space in K subsets is essential to the method, being responsible for providing diversity. The other important element is the rotation method used, because it influences the accuracy of the classifiers. In its original formulation, PCA was proposed to create the axes rotation on each subset of features, keeping all components in order to avoid losing discriminatory information. However, different transformations may be applied. In [11] PCA is compared to Nonparametric Discriminant Analysis [5], Sparse Random Projections and Random Projections [3]. PCA was found to be superior to the aforementioned alternatives. Recently, Independent Component Analysis [10] has been advocated as a preferred method for microarray gene expression problems. ICA looks for a set of features that are maximally independent for each other. In its linear version, it can be interpreted as finding the latent variables of the problem, whose linear combination models the observed data. Hence, it is sensible to apply ICA transformation to microarray gene expression, considering each extracted independent component as a potential biological process, like in [14], where ICA is found superior to PCA. Other authors also report on the advantages of using ICA as feature extraction method in microarray domain, for instance [16,28]. Hence, we consider both PCA and ICA, notating each corresponding version as RF-PCA and RF-ICA.

To compare Rotation Forest behavior with *classic* ensemble methods, Bagging [1] and Boosting [4] are also considered.

An important issue that strongly influences classifiers behavior on microarray data sets is the feature selection method. We have opted by SVM Recursive Feature Elimination (SVM-RFE), and ReliefF. SVM-RFE was introduced to gene selection in bioinformatics by Guyon et al. [8] and it is recognized as a very effective algorithm for microarray gene expression classification task [25,22]. ReliefF has been chosen because it is claimed [23] to behave well when few attributes are selected. Having two different attribute selection methods also allows checking whether the choice of RF-PCA or RF-ICA may be influenced by the attribute selection method.

3 Experimental Setting

Experiments were performed on 10 genomic and proteomic data sets, whose basic properties are summarized in Table 1. All data sets are publicly available at Kent Ridge Biomedical Data Set Repository [20].

Except for MLL and ALL, the data sets are related with binary classification problems. MLL is a three class problem, and ALL comprises seven different classes of pediatric acute lymphoblastic leukemia.

The combination of 2 attribute selection algorithms and 4 classification ensemble techniques produces 8 basic methods that are named joining their acronym by '+'. Hence, ReliefF+Bagging indicates that Bagging is applied to data sets filtered with ReliefF. Feature selection algorithms are invoked to obtain a specific number of features, given by an additional parameter. Since we are interested on the behavior of the methods when few attributes are selected, we have covered

Table 1. Data set description. All of them are free to download in [20].

| Data Set and original work reference | | Samples | Attributes | Items in each class |
|--------------------------------------|-------------------------|---------|------------|------------------------------------------------------------------------------------------|
| ALL-AML | Golub et al. (1999) | 72 | 7129 | {(47, 65.2%), (25, 34.8%)} |
| ALL | Yeoh et al. (2002) | 327 | 12558 | {(15, 4.6%), (27, 8.3%), (64, 19.6%), (20, 6.1%), (43, 13.1%), (79, 24.2%), (79, 24.2%)} |
| Breast | Van't Veer (2002) | 97 | 24481 | {(46, 47.4%), (51, 52.6%)} |
| CNS | Mukherjee et al. (2002) | 60 | 7129 | {(21, 35.0%), (39, 65.0%)} |
| Colon | Alon et al. (1999) | 62 | 2000 | {(22, 35.4%), (40, 64.6%)} |
| DLBCL | Alizadeh et al. (2000) | 47 | 4026 | {(24, 51.0%), (23, 49.0%)} |
| Lung | Gordon et al. (2002) | 181 | 12533 | {(31, 17.1%), (150, 82.9%)} |
| MLL | Armstrong et al. (2001) | 72 | 12582 | {(24, 33.3%), (20, 27.7%), (28, 38.8%)} |
| Ovarian | Petricoin et al. (2002) | 253 | 15154 | {(162, 64.0%), (91, 36.0%)} |
| Prostate | Singh et al. (2002) | 136 | 12600 | {(77, 56.6%), (59, 43.4%)} |

the range [4,128] in powers of 2. Consequently, we had a total of 48 ($2 \times 4 \times 6$) different configurations.

For each data set, error rates have been obtained with Nadeau and Bengio methodology that proposes the corrected resampled t-test for hypothesis testing [18]. This method requires 15 repetitions of training and test, where 90% randomly selected instances are used for training and the remaining 10% for testing. Feature selection techniques were applied internally to each training set to provide an honest evaluation.

All experiments were performed on the data mining framework Weka [26], with default values for all parameters, except in some cases: in SVM-RFE we set to 20% the number of features eliminated by the algorithm in each cycle; for all the classification ensemble techniques we used J48 pruned as the base classifiers and we set to 100 the number of base classifiers; in Rotation Forests we consider groups of 3 features (default value).

We have integrated a Java module of FastICA [13] as a filter in Weka. Thus, we can choose the axes projection method in Rotation Forest. The data are previously normalized.

Rankings and various statistical tests were performed with the publicly available software facilitated in [6].

4 Experimental Results

Table 2 summarizes the results obtained in our experiments, showing for each [method, number of selected attributes, data set] the average accuracy. We mark in **bold** the best results for each data set and selected number of attributes, and also we ordered the different methods always under the same schema.

Table 2. Summary of accuracy for all data sets, all methods and selected attributes

| Algorithm | #Attribs | ALL-AML | | | | | | | | | |
|--------------------|----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | ALL | Breast | CNS | Colon | DLBCL | Lung | MLL | Ovarian | prostate | |
| Relieff+Bagging | 4 | 95,32 | 72,48 | 65,41 | 57,37 | 79,84 | 88,56 | 97,41 | 82,74 | 96,31 | 87,22 |
| SVM-RFE+Bagging | 4 | 94,37 | 48,02 | 68,81 | 55,81 | 74,44 | 88,89 | 95,93 | 85,36 | 98,16 | 90,22 |
| Relieff+Boosting | 4 | 95,44 | 71,31 | 63,93 | 62,19 | 79,68 | 84,11 | 98,15 | 84,40 | 94,99 | 83,37 |
| SVM-RFE+Boosting | 4 | 92,54 | 45,81 | 61,93 | 54,70 | 70,48 | 86,44 | 96,30 | 86,07 | 98,16 | 93,22 |
| Relieff+RF-FastICA | 4 | 92,14 | 72,77 | 66,30 | 58,48 | 83,17 | 92,89 | 99,26 | 87,14 | 94,72 | 88,72 |
| SVM-RFE+RF-FastICA | 4 | 94,60 | 50,48 | 68,59 | 58,10 | 73,49 | 90,22 | 97,78 | 87,14 | 99,21 | 91,25 |
| Relieff+RF-PCA | 4 | 90,16 | 74,75 | 66,15 | 57,37 | 82,06 | 92,89 | 99,26 | 87,26 | 94,46 | 89,71 |
| SVM-RFE+RF-PCA | 4 | 93,77 | 50,87 | 65,41 | 56,03 | 74,60 | 93,00 | 97,78 | 84,29 | 100,0 | 90,77 |
| Relieff+Bagging | 8 | 91,19 | 80,17 | 65,41 | 61,65 | 75,56 | 90,44 | 97,78 | 88,10 | 96,31 | 90,18 |
| SVM-RFE+Bagging | 8 | 92,42 | 77,51 | 66,67 | 51,52 | 75,71 | 87,22 | 95,95 | 87,14 | 97,36 | 90,26 |
| Relieff+Boosting | 8 | 94,33 | 80,72 | 61,41 | 61,81 | 74,60 | 87,78 | 97,41 | 87,98 | 95,77 | 90,70 |
| SVM-RFE+Boosting | 8 | 94,88 | 79,35 | 63,41 | 53,43 | 75,56 | 84,78 | 95,56 | 90,00 | 97,90 | 93,66 |
| Relieff+RF-FastICA | 8 | 95,99 | 83,00 | 62,67 | 60,48 | 83,02 | 94,22 | 98,15 | 89,05 | 95,25 | 93,66 |
| SVM-RFE+RF-FastICA | 8 | 97,10 | 80,17 | 69,04 | 54,92 | 73,65 | 91,22 | 98,52 | 91,67 | 99,74 | 92,67 |
| Relieff+RF-PCA | 8 | 95,44 | 83,54 | 62,81 | 62,25 | 83,17 | 92,89 | 98,15 | 86,31 | 95,51 | 91,65 |
| SVM-RFE+RF-PCA | 8 | 98,21 | 80,19 | 65,33 | 54,60 | 77,78 | 92,33 | 98,15 | 89,88 | 100,0 | 90,22 |
| Relieff+Bagging | 16 | 92,54 | 84,40 | 65,33 | 55,14 | 78,73 | 91,56 | 97,06 | 88,93 | 97,10 | 92,67 |
| SVM-RFE+Bagging | 16 | 93,37 | 85,11 | 68,67 | 54,70 | 76,83 | 88,56 | 95,58 | 83,45 | 97,10 | 87,25 |
| Relieff+Boosting | 16 | 94,33 | 86,23 | 62,89 | 60,54 | 77,94 | 80,11 | 97,80 | 86,19 | 96,84 | 92,60 |
| SVM-RFE+Boosting | 16 | 91,27 | 87,65 | 62,07 | 56,92 | 73,49 | 87,78 | 94,83 | 87,26 | 97,91 | 94,69 |
| Relieff+RF-FastICA | 16 | 97,10 | 87,44 | 64,15 | 62,92 | 85,24 | 94,56 | 98,89 | 92,62 | 97,36 | 93,63 |
| SVM-RFE+RF-FastICA | 16 | 98,21 | 87,61 | 66,30 | 60,70 | 77,94 | 91,56 | 98,52 | 89,05 | 99,74 | 89,27 |
| Relieff+RF-PCA | 16 | 95,16 | 87,03 | 62,81 | 57,21 | 84,13 | 94,56 | 98,52 | 90,83 | 98,41 | 92,60 |
| SVM-RFE+RF-PCA | 16 | 98,21 | 88,07 | 66,89 | 59,43 | 80,00 | 94,00 | 98,15 | 89,05 | 99,73 | 92,67 |
| Relieff+Bagging | 32 | 92,54 | 84,86 | 61,93 | 51,81 | 77,94 | 92,89 | 97,06 | 90,83 | 97,36 | 91,68 |
| SVM-RFE+Bagging | 32 | 93,25 | 86,99 | 63,41 | 56,92 | 81,11 | 90,22 | 96,32 | 87,86 | 97,11 | 87,21 |
| Relieff+Boosting | 32 | 91,83 | 87,47 | 62,15 | 53,87 | 77,94 | 81,11 | 98,17 | 90,95 | 98,15 | 93,59 |
| SVM-RFE+Boosting | 32 | 91,43 | 91,87 | 64,30 | 65,21 | 77,94 | 86,11 | 94,83 | 89,88 | 98,16 | 95,16 |
| Relieff+RF-FastICA | 32 | 97,10 | 88,71 | 63,48 | 56,25 | 85,24 | 95,89 | 98,15 | 91,79 | 98,93 | 95,09 |
| SVM-RFE+RF-FastICA | 32 | 97,10 | 90,83 | 67,56 | 63,87 | 76,98 | 94,22 | 98,52 | 92,62 | 99,48 | 91,65 |
| Relieff+RF-PCA | 32 | 98,21 | 88,27 | 64,96 | 58,48 | 83,17 | 95,89 | 98,15 | 90,83 | 99,47 | 92,64 |
| SVM-RFE+RF-PCA | 32 | 98,21 | 90,85 | 72,44 | 70,98 | 80,32 | 93,11 | 98,52 | 89,88 | 100,0 | 93,11 |
| Relieff+Bagging | 64 | 93,37 | 88,08 | 61,19 | 49,37 | 79,84 | 92,89 | 97,06 | 89,88 | 97,11 | 89,19 |
| SVM-RFE+Bagging | 64 | 95,32 | 87,43 | 67,04 | 64,70 | 77,94 | 91,56 | 95,95 | 86,90 | 97,11 | 86,66 |
| Relieff+Boosting | 64 | 94,44 | 89,71 | 64,74 | 58,10 | 79,05 | 80,44 | 98,91 | 91,79 | 97,36 | 92,60 |
| SVM-RFE+Boosting | 64 | 92,22 | 91,69 | 60,52 | 64,03 | 77,78 | 87,11 | 96,32 | 91,79 | 97,64 | 91,61 |
| Relieff+RF-FastICA | 64 | 97,10 | 89,91 | 61,33 | 64,60 | 83,17 | 95,89 | 98,52 | 91,79 | 98,94 | 94,14 |
| SVM-RFE+RF-FastICA | 64 | 97,10 | 91,51 | 66,22 | 68,32 | 78,10 | 92,89 | 98,15 | 92,62 | 99,73 | 93,15 |
| Relieff+RF-PCA | 64 | 97,10 | 89,91 | 63,33 | 63,87 | 85,24 | 94,56 | 98,52 | 90,00 | 99,73 | 93,15 |
| SVM-RFE+RF-PCA | 64 | 97,10 | 92,11 | 67,63 | 67,21 | 83,17 | 93,22 | 98,15 | 90,83 | 100,0 | 94,65 |
| Relieff+Bagging | 128 | 93,37 | 88,90 | 58,96 | 56,19 | 81,11 | 92,89 | 96,67 | 91,79 | 96,84 | 89,67 |
| SVM-RFE+Bagging | 128 | 93,49 | 87,84 | 61,93 | 66,92 | 79,84 | 92,89 | 96,32 | 86,07 | 97,11 | 86,74 |
| Relieff+Boosting | 128 | 94,33 | 90,70 | 57,93 | 53,90 | 80,95 | 81,44 | 98,15 | 92,74 | 97,36 | 91,68 |
| SVM-RFE+Boosting | 128 | 94,17 | 91,88 | 64,07 | 64,92 | 83,02 | 83,11 | 97,06 | 89,76 | 97,90 | 87,17 |
| Relieff+RF-FastICA | 128 | 97,10 | 90,52 | 62,07 | 57,52 | 85,24 | 95,89 | 98,89 | 95,48 | 99,73 | 94,14 |
| SVM-RFE+RF-FastICA | 128 | 98,06 | 91,50 | 64,15 | 68,63 | 81,11 | 94,22 | 98,52 | 93,57 | 99,73 | 93,15 |
| Relieff+RF-PCA | 128 | 97,10 | 90,91 | 64,00 | 59,75 | 85,24 | 95,89 | 98,89 | 92,74 | 99,73 | 93,63 |
| SVM-RFE+RF-PCA | 128 | 98,21 | 91,28 | 68,15 | 63,71 | 83,17 | 94,22 | 98,52 | 94,52 | 100,0 | 95,05 |

We can observe the following behavior. As a general rule, the accuracy increases as the number of attributes increases. With a very low number of attributes, there is no specific method that clearly performs better than the rest. And there is a clear trend in favor of Rotation Forests (either PCA or FastICA) when the number of selected attributes increases.

Table 3. Ranking of the methods using the significant differences from all pairwise comparisons. "W" stands for "wins" and "L" for "losses".

| Method | W-L | W-L | W-L | W-L | W-L | W-L | Total |
|--------------------|-----|-----|-----|-----|-----|-----|-----------|
| # attributes | 4 | 8 | 16 | 32 | 64 | 128 | Dominance |
| SVM-RFE+RF-PCA | 0 | 4 | 3 | 8 | 5 | 3 | 23 |
| SVM-RFE+RF-FastICA | 0 | 3 | 1 | 2 | 2 | 5 | 13 |
| ReliefF+RF-PCA | 0 | -2 | 2 | 0 | 2 | 3 | 5 |
| ReliefF+RF-FastICA | 1 | -4 | 2 | 0 | 3 | 2 | 4 |
| SVM-RFE+Boosting | -1 | 2 | -3 | 1 | 0 | 1 | 0 |
| ReliefF+Boosting | 0 | -1 | -2 | -1 | -1 | -3 | -8 |
| ReliefF+Bagging | 2 | -2 | 0 | -4 | -6 | -4 | -14 |
| SVM-RFE+Bagging | -2 | 0 | -3 | -6 | -5 | -7 | -23 |

Table 3 shows a dominance ranking of the methods according to the difference between the number of times each method has been significantly better and significantly worse than another method, considering all pairwise comparisons for a given number of genes. Last column shows the total dominance, obtained adding the results for each considered number of attributes.

We can see that selecting four attributes, only Bagging and Rotation Forest with FastICA, both using ReliefF, have a difference (Wins-Losses) > 0; selecting eight attributes, both Rotation Forest (with PCA and FastICA) and Boosting have that difference > 0. Increasing the number of attributes selected, Rotation Forest with PCA and FastICA achieve to improve those differences whereas in the other methods they get worse.

According to Table 3, the best results are obtained using Rotation Forest with PCA or FastICA and SVM-RFE, being PCA better than FastICA. The following methods in the ranking are both Rotation Forest with ReliefF.

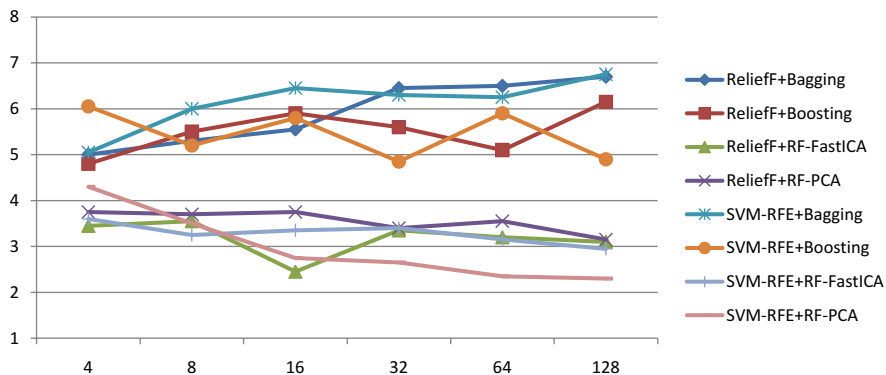


Fig. 1. Rankings for all the algorithms. In horizontal is represented the number of selected attributes. In vertical the average ranking position.

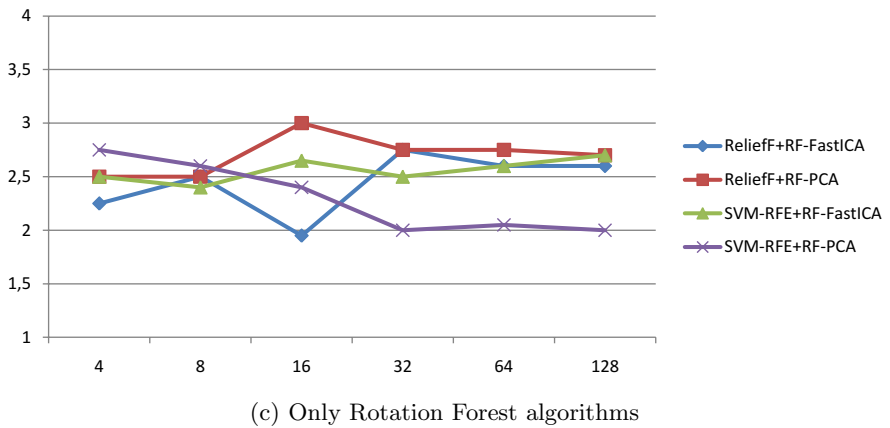
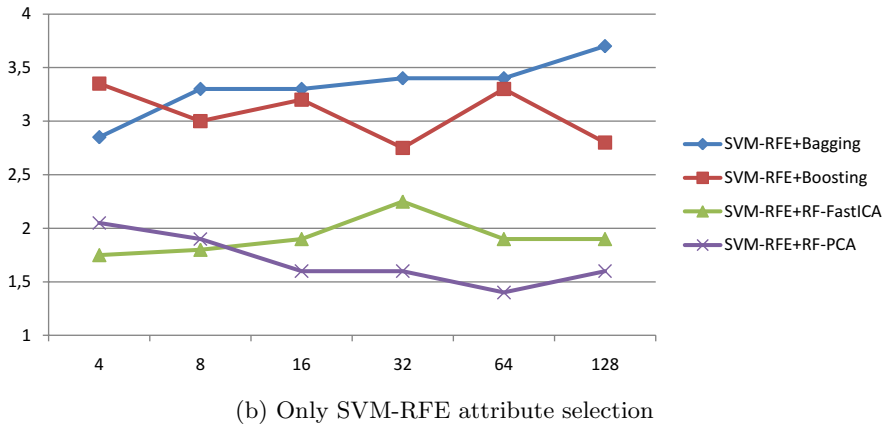
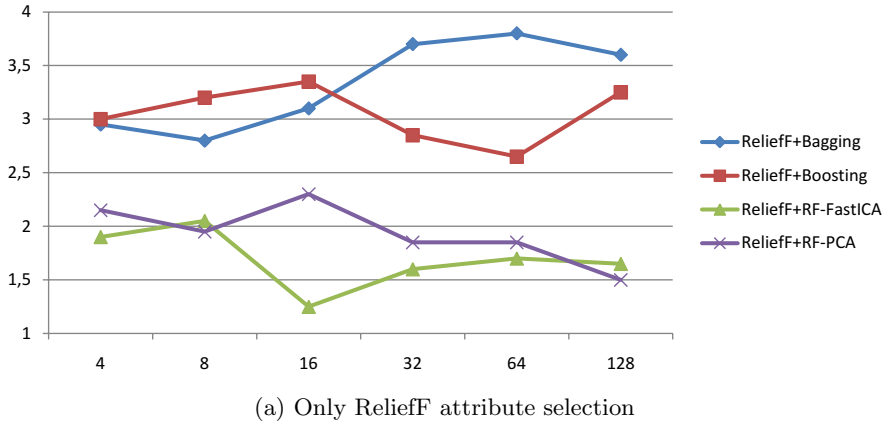


Fig. 2. Various rankings

When several data sets are involved, using only the average accuracy sometimes blurs the effect of the different algorithms. Hence, we decided to use rankings to establish which algorithms are the most successful in a global view [2]. Using the freely available software [6], we obtain some statistics and various rankings that are shown in next figures.

Figure 1 shows clearly two different branches: one containing the classic ensemble methods (*Bagging* and *Boosting*) on the upper part of the figure, and other branch containing those algorithms using *Rotation Forest* with *PCA* and *FastICA* as projection techniques. Notice that the lowest ranking positions are related with better performing algorithms.

Iman-Davenport's test rejects the null hypothesis (all methods are equivalent on their ranks) for all the algorithms working with more than 4 selected attributes.

We perform a study for each number of selected attributes using *Bonferroni-Dum's* procedure, which compares the best algorithm against the rest. The results show that: a) For more than 8 selected attributes, the null hypothesis can be rejected for *Bagging* in all the attribute selection methods used. b) For *Boosting* similar conclusions can be drawn but limited to ReliefF algorithm, and c) For very low number of selected attributes (less than 16), none of the methods can be rejected under the null hypothesis, except Bagging with SVM-RFE and 8 attributes.

We studied also the behavior of the different classifying techniques in relation with the attribute selection method used.

The rankings showed in figure 2 can be interpreted in the same way as we have done earlier. Subfigures (a) and (b) show clearly that Rotation Forests outperforms the classic ensemble methods, and also that only for a low number of selected attributes and ReliefF, the null hypothesis cannot be rejected.

In figure 2(c) can be seen a study of the behavior of Rotation Forest, comparing only the performance of the two projection techniques used in this work. In this case, there are no strong conclusions to be drawn, and it can be said that statistically speaking, all the algorithms involving Rotation Forests are equivalent, that is, the null hypothesis about equal rank position cannot be rejected. But by mere inspection of the graphic, it can be seen that SVM-RFE as attribute selection technique and Rotation Forest with PCA produces usually better results when the number of attributes is greater than 16; whereas ReliefF with Rotation Forests and FastICA are preferred for low attribute number.

To consider the effect of pruning, experiments have also been performed combining unpruned trees. Experimental results are quite similar, allowing to extract the same conclusions.

5 Conclusions

Experimental work has shown that using Independent Component Analysis as a projection method in Rotation Forest does not generally improve on Rotation Forest with Principal Component Analysis. Moreover, according to dominance ranks, Rotation Forest PCA is the preferred method.

According to Friedman rankings, for 16 or more attributes, Rotation Forest, both PCA and ICA, outperforms Bagging and Boosting, but differences blur for 8 and especially 4 attributes using ReliefF as attribute selection method.

A direct comparison of Rotation Forest methods does not find statistically significant differences on their rankings. This conclusion differs from previously found results for some specific data sets. Several issues may influence the behavior of the algorithms, such as data nature, attribute selection technique, and Rotation Forest parameters and projection method. Further research is needed to properly understand the influence of the projection method on Rotation Forest behavior.

Acknowledgments. This work has been partially funded by Junta de Castilla y León through grant VA100A08.

References

1. Breiman, L.: Bagging predictors. *Machine Learning* 24(2), 123–140 (1996)
2. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
3. Fern, X.Z., Broadley, C.E.: Random projection for high dimensional data clustering: A cluster ensemble approach. In: *Proc. 20th International Conference on Machine Learning, ICML*, pp. 186–193 (2003)
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Computer and System Sciences* 55(1), 119–139 (1997)
5. Fukunaga, K., Mantock, J.: Nonparametric discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5(3), 671–678 (1983)
6. Garcia, S., Herrera, F.: An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research* 9, 2677–2694 (2008)
7. Golub, T.R., Stomin, D.K., Tamayo, P.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537 (1999)
8. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 389–422 (2002)
9. Han, J., Kanber, M.: *Data Mining: Concepts and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2006)
10. Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and applications. *Neural Networks* 14(4-5), 411–430 (2000)
11. Kuncheva, L.I., Rodríguez, J.J.: An experimental study on rotation forest ensembles. In: Haindl, M., Kittler, J., Roli, F. (eds.) *MCS 2007. LNCS*, vol. 4472, pp. 459–468. Springer, Heidelberg (2007)
12. Kuncheva, L.I.: *Combining Pattern Classifiers: Methods and Algorithms*. Wiley Interscience, Hoboken (2004)
13. Lambertz, M.: *Fastica for java* (2006), <http://sourceforge.net/projects/fastica/>
14. Lee, S., Batzoglou, S.: Application of independent component analysis to microarrays. *Genome Biology* 4(11) (2003)

15. Li, W., Yang, Y.: How many genes are needed for a discriminant microarray data analysis? In: Critical Assessment of Techniques for Microarray Data Mining Workshop, pp. 137–150 (2000)
16. Liebermeister, W.: Linear modes of gene expressions determined by independent component analysis. *Bioinformatics* 18, 51–56 (2002)
17. Liu, K., Huang, D.: Cancer classification using rotation forest. *Computers in Biology and Medicine* 38, 601–610 (2008)
18. Nadeau, C., Bengio, Y.: Inference for the generalization error. *Machine Learning* 52(3), 239–281 (2003)
19. Nanni, L., Lumini, A.: Using ensemble of classifiers in Bioinformatics. In: Machine Learning Research Progress. Nova Science publisher (2009)
20. Ridge, K.: Kent ridge bio-medical dataset (2009), <http://datam.i2r.a-star.edu.sg/datasets/krbd/>
21. Rodríguez, J.J., Kuncheva, L.I., Alonso-González, C.J.: Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(10), 1619–1621 (2006)
22. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. *Bioinformatics* 23, 2507–2517 (2007)
23. Stiglic, G., Rodríguez, J.-J., Kokol, P.: Feature selection and classification for small gene sets. In: Chetty, M., Ngom, A., Ahmad, S. (eds.) *PRIB 2008. LNCS (LNBI)*, vol. 5265, pp. 121–131. Springer, Heidelberg (2008)
24. Symons, S., Nieselt, K.: Data mining microarray data - Comprehensive benchmarking of feature selection and classification methods, Pre-print, www.zbit.uni-tuebingen.de/pas/preprints/GCB2006/SymonsNieselt.pdf
25. Tang, Y., Zhang, Y., Huang, Z.: FCM-SVM-RFE gene feature selection algorithm for leukemia classification from microarray gene expression data. In: *FUZZ 2005, The 14th IEEE International Conference on Fuzzy Systems*, pp. 97–101 (2005)
26. Witten, I., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)
27. Xiong, M., Fang, Z., Zhao, J.: Biomarker identification by feature wrappers. *Genome Research* 11, 1878–1887 (2001)
28. Zhang, X.W., Yap, Y.L., Wei, D., Chen, F., Danchin, A.: Molecular diagnosis of human cancer type by gene expression profiles and independent component analysis. *European J. Human Genetics* 13, 1303–1311 (2005)

An Empirical Study of Multilayer Perceptron Ensembles for Regression Tasks*

Carlos Pardo, Juan José Rodríguez, César García-Osorio, and Jesús Maudes

University of Burgos, Spain

{cpardo, jjrodriguez, jmaudes, cgosorio}@ubu.es

Abstract. This work presents an experimental study of ensemble methods for regression, using Multilayer Perceptrons (MLP) as the base method and 61 datasets. The considered ensemble methods are Randomization, Random Subspaces, Bagging, Iterated Bagging and AdaBoost.R2. Surprisingly, because it is in contradiction to previous studies, the best overall results are for Bagging. The cause of this difference can be the base methods, MLP instead of regression or model trees. Diversity-error diagrams are used to analyze the behaviour of the ensemble methods. Compared to Bagging, the additional diversity obtained with other methods do not compensate the increase in the errors of the ensemble members.

1 Introduction

Ensembles [1] are combinations of models. In many situations, an ensemble gives better results than any of its members. Although they have been studied mainly for classification, there are also ensemble methods for regression.

The models to be combined have to be different, otherwise the ensemble is unnecessary. One way to have different models is to construct them with different methods. Nevertheless, there are ensemble methods that combine models obtained from the same method. Most of these ensemble methods change the dataset in some way.

In Bagging [2] each member is trained with a sample of the training data. Normally, the size of the sample is the same than the size of the original training data, but the sample is *with replacement*. Hence, some training examples will appear several times in the sample while others will not appear. The prediction of the ensemble is the average of its members predictions.

In Random Subspaces [3] each member is trained with all the training examples, but with a subset of the attributes. The dimension of the subspaces is a parameter of the method. The prediction of the ensemble is also the average of the predictions.

Bagging and Random Subspaces can be used for classification and for regression. AdaBoost [4] initially was a method for classification, but there are some variants for regression, such as AdaBoost.R2 [5]. In these methods, each training example has a weight. Initially, all the examples have the same weight. The construction of the ensemble members must take into account the examples weights. After an ensemble member is constructed, the examples weights are adjusted. The idea is to give more weight to

* This work was supported by the Project 2009/00204/001 of “Caja de Burgos” and University of Burgos and the Project TIN2008-03151 of the Spanish Ministry of Education and Science.

the examples with greater errors in the previous iterations. Hence, in the construction of the next member, these examples will be more important. The ensemble members also have weights, they depend on their error. In AdaBoost.R2, the predicted value of the ensemble is a weighted median. In [6], this method was the one with the best results among several ensemble methods for regression.

Iterated Bagging [7] is a method for regression based on Bagging. It combines several Bagging ensembles. The first Bagging ensemble is constructed as usual. Based on the predictions of the previous Bagging ensemble, the values of the predicted variable are altered. The next Bagging ensemble is trained with these altered values. These values are the *residuals*: the difference between the real and the predicted values. Nevertheless, these predictions are not obtained using all the members in the Bagging ensemble. The error of the predictions for a training example would be too optimistic, the majority of the ensemble methods have been trained with that example. These predictions are obtained using the *out-of-bag* estimation: the prediction for an example is obtained using only those ensemble members that were not trained with that example. The prediction of an Iterated Bagging ensemble is the sum of the predictions of its Bagging ensembles. According to [8], Iterated Bagging is in general the most effective method.

In Negative Correlation Learning [9], the ensemble members are constructed in parallel. The networks are trained using a penalty term, they penalize the similarity of the current network with the ensemble. This method has not been considered in this work because it requires the use of a modified base method.

The rest of the paper is organised as follows. Next section details the experimental settings. The results are discussed in section 3. Section 4 is dedicated to diversity error diagrams. Finally, section 5 presents some concluding remarks.

2 Experimental Settings

The experiments were conducted using 5×2 fold cross validation [10]. The performance of the different methods over different datasets was measured using *root mean squared error* (RMSE). The base models were Multilayer Perceptrons. Ensemble size was 50.

Several ensemble methods were considered:

- Randomization. When the base method has a random element, different models can be obtained from the same training data. In the case of Multilayer Perceptrons, the initial weights are initialized randomly. Randomization is an ensemble of such randomizable models, which prediction is the average of the members predictions.
- Bagging [2].
- Random Subspaces [3]. For the dimension of the subspaces, two values were considered: 50% and 75% of the number of attributes.
- AdaBoost.R2 [5]. This method can be used with different loss functions. Three are proposed in [5] and used in this work: linear, square and exponential. The suffixes “-Li”, “-Sq” and “-Ex” are used to denote the used function. Moreover, methods based on AdaBoost can be used in two ways [11]. In the reweighting version, the base model is trained with all the training data, it must take into account the weight distribution. In the resampling version, the base model is trained with a sample

from the training data. This sample is constructed taken into account the weights. These versions are denoted with “-W” and “-S”.

- Iterated Bagging [7]. The used configuration was 5×10 : Bagging is iterated 5 times, the ensemble size of each Bagging is 10.

Moreover, other methods were included in the study, as a baseline for the comparisons:

- A single MLP.
- Linear regression. Two versions were considered: using all the features and using only the selected features with the method described in [12].
- Nearest neighbors. There are two versions, in the first one the number of neighbors is 1. In the other, the number of neighbors is selected using “leave one out”.

Weka [13] was used for the experiments. It includes the base method (Multilayer Perceptron), Bagging and Random Subspaces. The rest of the methods (i.e., Iterated Bagging and AdaBoost.R2), were implemented in this library.

For Multilayer Perceptrons the default settings in Weka were used. There is a hidden layer, the number of neurons is half the number of attributes (including the objective attribute). The learning rate is 0.3, the momentum is 0.2 and the number of epochs is 500.

Table 1 shows the characteristics of the 61 considered datasets. They are available in the format used by Weka [1]. 30 of them were collected by Luis Torgo [2].

3 Results

In order to compare all the configurations considered, average ranks [14] were used. For each dataset, the methods are sorted according to their performance. The best method has rank 1, the second rank 2 and so on. If there are ties, these methods have the same rank, the average value. For each method, its average rank is obtained as the average value over all the considered datasets. According to [14], “*average ranks by themselves provide a fair comparison of the algorithms*”. Table 2 shows the methods sorted according to their average ranks.

A single MLP has worse average rank than k-Nearest Neighbors and Linear Regression. Moreover, an ensemble of MLP based only in the random initialization of weights is also worse than these baseline methods. Nevertheless, several ensembles of MLP have better average ranks than the baseline methods.

The best average rank is for Bagging. The second method is Random Subspaces, using 75% of the attributes. The next positions are for AdaBoost.R2, the version that uses resampling. On the other hand, the results for AdaBoost.R2 are among the worst when training the base models directly with the weighted instances.

Table 3 shows a direct comparison of a single MLP and Bagging with the other methods. When comparing two methods, the number of datasets where one method has better, equal, or worse results than the other is calculated. According to [14], using a sign test, one method is significantly better than other, with a confidence level of 0.05, if the number of wins plus half the ties is at least $N/2 + 1.96\sqrt{N}/2$. For $N = 61$ datasets, this number is 39.

¹ http://www.cs.waikato.ac.nz/ml/weka/index_datasets.html

² <http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html>

Table 1. Datasets used in the experiments

| Dataset | Examples | Numeric | Nominal | Dataset | Examples | Numeric | Nominal |
|------------------|----------|---------|---------|-------------|----------|---------|---------|
| 2d-planes | 40768 | 10 | 0 | house-16H | 22784 | 16 | 0 |
| abalone | 4177 | 7 | 1 | house-8L | 22784 | 8 | 0 |
| aileron | 13750 | 40 | 0 | housing | 506 | 12 | 1 |
| auto93 | 93 | 16 | 6 | hungarian | 294 | 6 | 7 |
| auto-horse | 205 | 17 | 8 | kin8nm | 8192 | 8 | 0 |
| auto-mpg | 398 | 4 | 3 | longley | 16 | 6 | 0 |
| auto-price | 159 | 15 | 0 | lowbwt | 189 | 2 | 7 |
| bank-32nh | 8192 | 32 | 0 | machine-cpu | 209 | 6 | 0 |
| bank-8FM | 8192 | 8 | 0 | mbgrade | 61 | 1 | 1 |
| basketball | 96 | 4 | 0 | meta | 528 | 19 | 2 |
| bodyfat | 252 | 14 | 0 | mv | 40768 | 7 | 3 |
| bolts | 40 | 7 | 0 | pbcc | 418 | 10 | 8 |
| breast-tumor | 286 | 1 | 8 | pharynx | 195 | 1 | 10 |
| cal-housing | 20640 | 8 | 0 | pole | 15000 | 48 | 0 |
| cholesterol | 303 | 6 | 7 | pollution | 60 | 15 | 0 |
| cleveland | 303 | 6 | 7 | puma32H | 8192 | 32 | 0 |
| cloud | 108 | 4 | 2 | puma8NH | 8192 | 8 | 0 |
| cpu-act | 8192 | 21 | 0 | pw-linear | 200 | 10 | 0 |
| cpu | 209 | 6 | 1 | pyrimidines | 74 | 27 | 0 |
| cpu-small | 8192 | 12 | 0 | quake | 2178 | 3 | 0 |
| delta-aileron | 7129 | 5 | 0 | schlvote | 38 | 4 | 1 |
| delta-elevators | 9517 | 6 | 0 | sensory | 576 | 0 | 11 |
| detroit | 13 | 13 | 0 | servo | 167 | 0 | 4 |
| diabetes-numeric | 43 | 2 | 0 | sleep | 62 | 7 | 0 |
| echo-months | 130 | 6 | 3 | stock | 950 | 9 | 0 |
| elevators | 16599 | 18 | 0 | strike | 625 | 5 | 1 |
| elusage | 55 | 1 | 1 | triazines | 186 | 60 | 0 |
| fishcatch | 158 | 5 | 2 | veteran | 137 | 3 | 4 |
| friedman | 40768 | 10 | 0 | vineyard | 52 | 3 | 0 |
| fruitfly | 125 | 2 | 2 | wisconsin | 194 | 32 | 0 |
| gascons | 27 | 4 | 0 | | | | |

The number of wins, ties and losses and the average ranks are calculated using a direct comparison of the results for the different methods. Nevertheless, they do not take into account the size of the differences. For this purpose, we use the *quantitative scoring* [15,6]. Given the results for two methods i and j in one dataset, this score is defined as

$$S_{i,j} = \frac{RMSE_j - RMSE_i}{\max(RMSE_i, RMSE_j)}$$

Where $RMSE_i$ is the root mean squared error for the method i . Unless both methods have zero error, this measure will be between -1 and 1 , although it can be expressed as a percentage. The sign indicates which method is better.

Figure 1 shows these scores (as percentages) for the considered methods, compared with Bagging. The score is calculated for each dataset and the datasets are sorted

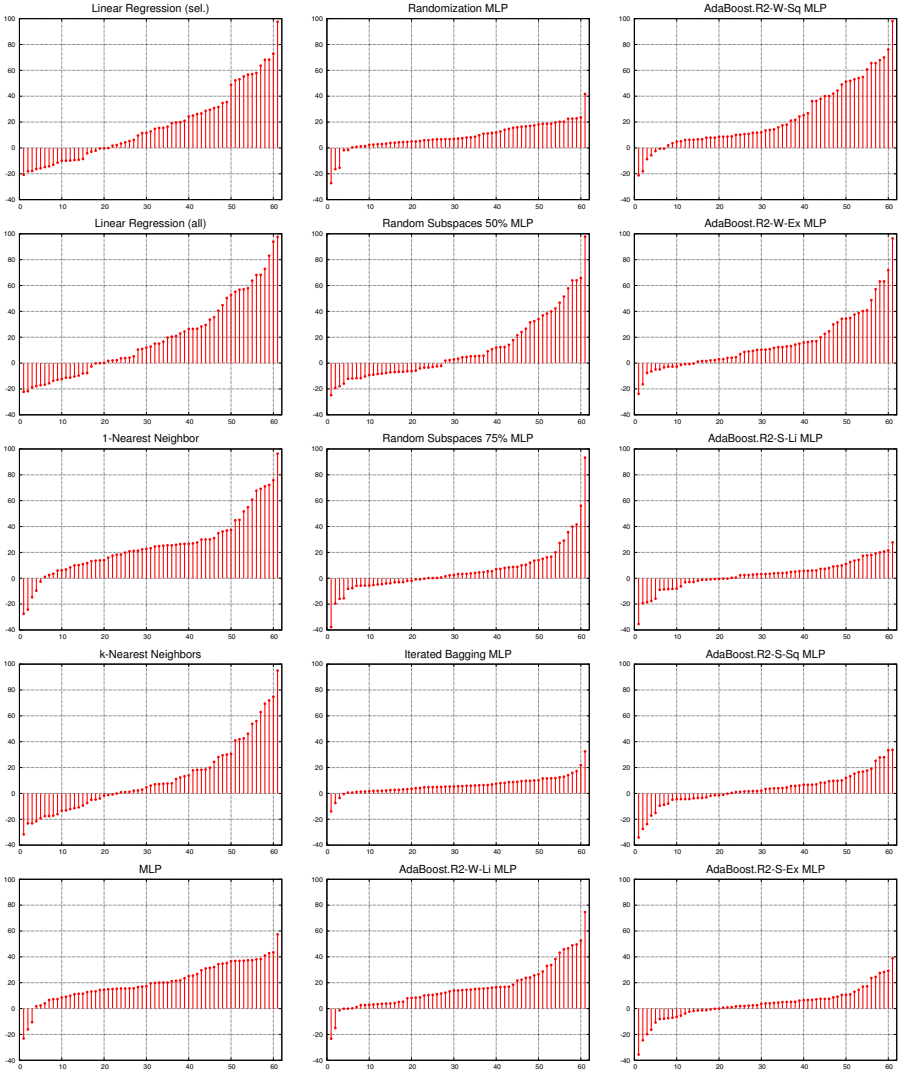


Fig. 1. Comparison scores between the considered methods and Bagging of Multilayer Perceptrons

according to its score. The number of values above and below zero corresponds with the number of wins and losses in Table 3.

When comparing two methods with these graphs, it is desired to have more positive values than negative, but it is also desirable that the absolute values were greater for positive scores than for negative scores. In this case there are more positive values, and the greater absolute scores are also for positive values.

The results for Iterated Bagging are clearly worse than the results for Bagging. This can be caused by the selected configuration, 5 iterations of Bagging with 10 ensembles.

Table 2. Considered methods sorted according to their average ranks

| | Average rank | Method |
|----|--------------|-------------------------------|
| 1 | 4.61 | Bagging MLP |
| 2 | 6.28 | Random Subspaces 75% MLP |
| 3 | 6.74 | AdaBoost.R2-S-Li MLP |
| 4 | 7.12 | AdaBoost.R2-S-Sq MLP |
| 5 | 7.21 | AdaBoost.R2-S-Ex MLP |
| 6 | 7.30 | Random Subspaces 50% MLP |
| 7 | 7.35 | k-Nearest Neighbors |
| 8 | 7.37 | Iterated Bagging MLP |
| 9 | 8.42 | Linear Regression (all) |
| 10 | 8.55 | Linear Regression (selection) |
| 11 | 8.63 | Randomization MLP |
| 12 | 9.48 | AdaBoost.R2-W-Ex MLP |
| 13 | 10.69 | AdaBoost.R2-W-Li MLP |
| 14 | 11.82 | Single MLP |
| 15 | 11.82 | AdaBoost.R2-W-Sq MLP |
| 16 | 12.62 | 1-Nearest Neighbor |

Table 3. Comparison of ensembles of Multilayer Perceptrons with a single model and Bagging. The number of wins, ties and losses is shown for the comparison of the column method with the row method.

| Method | Single MLP | Bagging |
|-------------------------------|--------------------|--------------------|
| Linear Regression (selection) | 36 / 0 / 25 | 21 / 0 / 40 |
| Linear Regression (all) | 34 / 0 / 27 | 19 / 0 / 42 |
| 1-Nearest Neighbor | 23 / 0 / 38 | 5 / 0 / 56 |
| k-Nearest Neighbors | 42 / 0 / 19 | 23 / 0 / 38 |
| Single MLP | 0 / 61 / 0 | 3 / 0 / 58 |
| Randomization MLP | 60 / 0 / 1 | 5 / 0 / 56 |
| Bagging MLP | 58 / 0 / 3 | 0 / 61 / 0 |
| Random Subspaces 50% MLP | 39 / 1 / 21 | 27 / 0 / 34 |
| Random Subspaces 75% MLP | 52 / 0 / 9 | 23 / 0 / 38 |
| Iterated Bagging MLP | 54 / 0 / 7 | 4 / 0 / 57 |
| AdaBoost.R2-W-Li MLP | 39 / 0 / 22 | 5 / 0 / 56 |
| AdaBoost.R2-W-Sq MLP | 35 / 0 / 26 | 7 / 0 / 54 |
| AdaBoost.R2-W-Ex MLP | 41 / 0 / 20 | 14 / 0 / 47 |
| AdaBoost.R2-S-Li MLP | 48 / 1 / 12 | 22 / 0 / 39 |
| AdaBoost.R2-S-Sq MLP | 48 / 0 / 13 | 22 / 0 / 39 |
| AdaBoost.R2-S-Ex MLP | 50 / 0 / 11 | 20 / 0 / 41 |

That configuration was selected because it was desired to compare ensembles with the same number of base models, 50. That is, the Bagging configuration (10 base models) that is iterated, is not the same that the Bagging configuration (50 base models) that is not iterated. If more base models were allowed, Iterated Bagging could be used with Bagging of 50 models and it could improve the results of Bagging.

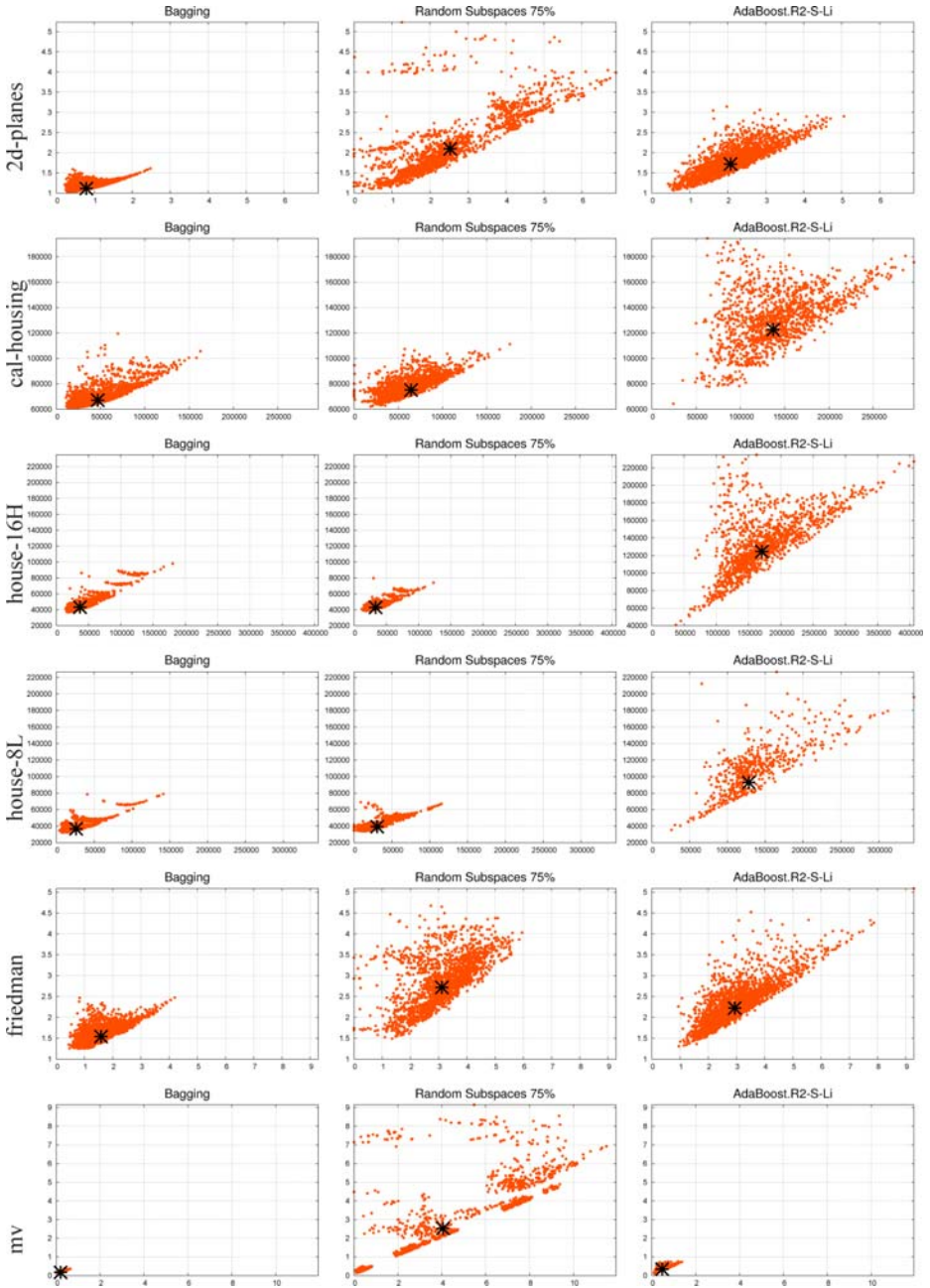


Fig. 2. Diversity error diagrams

4 Diversity-Error Diagrams

Successful ensembles are formed by models with low errors, but that are diverse. These two objectives are contradictory, because if the errors of two models are small, they cannot be very different. Several diversity measures had been proposed in order to analyze the behaviour of ensemble methods [16].

One of the techniques used is diversity-error diagrams [17]. They are scatter plots, there is a point for each pair of models. The horizontal axis represents the diversity between the two models, for classification, usually κ (kappa) is used. The vertical axis represents the average error of the two models.

In regression, several error measures can be considered, in this work RMSE was used:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (a_i - p_i)^2}{n}}$$

Where a_i are the actual values and p_i are the predicted values.

For measuring the diversity, the RMSE of one of the models with respect to the other was used:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (q_i - p_i)^2}{n}}$$

Where p_i and q_i are the predictions of the two models. Note that with this measure, bigger values indicate more diversity, while for kappa, bigger values indicated less diversity.

Figure 2 shows these diagrams for the best three methods according to the average ranks and the datasets with more instances. In general, the ensemble members for Bagging have the smallest errors, but they are the less diverse. In this case, it seems that the additional diversity in Random Subspaces and AdaBoost.R2 does not compensate the increased errors of the ensemble members.

5 Conclusions

The performance of ensemble methods for regression have been studied, using Multilayer Perceptrons as base method. The considered ensemble methods have been Randomization, Random Subspaces, Bagging, Iterated Bagging and AdaBoost.R2.

The best method, according to the average ranks, is Bagging, followed by Random Subspaces (being the subspace size 75% of the original space). These results disagree with previous studies [8,6,18], where Iterated Bagging or AdaBoost.R2 had better results. One source of the differences can be the different datasets used, but in [18] the considered datasets were the same than for this work. The difference in the results can be caused by the different base models considered, Multilayer Perceptrons instead or regression or model trees.

These results were obtained with predefined settings: the ensemble size (50), the default parameters for the Multilayer Perceptron, the subspace sizes (50% or 75%) for the Random Subspaces method, ... Different settings could give different conclusions.

The values of some parameters could be adjusted, although increasing significantly the computation time. Nevertheless, the comparison is fair because the same base classifier method was used and the ensembles were formed by the same number of base classifier.

Diversity-error diagrams have been used to study the behavior of the ensemble members. In this case, it seems to be more important to have member with low error than high diversity, because this is the situation for the best method, Bagging.

The best results are obtained with one of the more simple ensemble methods, Bagging, hence an open question is if there are other ensemble methods more suited for combining Multilayer Perceptrons in regression tasks.

Given the disparity of the results for the ensemble methods when using trees and Multilayer Perceptrons as base models, it seems interesting to study the behavior of ensemble methods using other base models, specially other types of Neural Networks, such as RBF (Radial Basis Function) Networks.

Acknowledgements. We wish to thank the developers of Weka. We also express our gratitude to the donors of the different datasets.

References

1. Kuncheva, L.I.: *Combining Pattern Classifiers: Methods and Algorithms*. Wiley Interscience, Hoboken (2004)
2. Breiman, L.: Bagging predictors. *Machine Learning* 24(2), 123–140 (1996)
3. Ho, T.K.: The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8), 832–844 (1998)
4. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: *13th International Conference on Machine Learning*, pp. 148–156. Morgan Kaufmann, San Francisco (1996)
5. Drucker, H.: Improving regressors using boosting techniques. In: *ICML 1997: Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 107–115. Morgan Kaufmann Publishers Inc., San Francisco (1997)
6. Zhang, C., Zhang, J., Wang, G.: An empirical study of using rotation forest to improve regressors. *Applied Mathematics and Computation* 195(2), 618–629 (2008)
7. Breiman, L.: Using iterated bagging to debias regressions. *Machine Learning* 45(3), 261–277 (2001)
8. Suen, Y., Melville, P., Mooney, R.: Combining bias and variance reduction techniques for regression trees. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) *ECML 2005. LNCS (LNAI)*, vol. 3720, pp. 741–749. Springer, Heidelberg (2005)
9. Brown, G., Wyatt, J.L., Tiño, P.: Managing diversity in regression ensembles. *Journal of Machine Learning Research* 6, 1621–1650 (2005)
10. Dietterich, T.G.: Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation* 10(7), 1895–1923 (1998)
11. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1), 119–139 (1997)
12. Wang, Y., Witten, I.H.: Induction of model trees for predicting continuous classes. In: *Poster papers of the 9th European Conference on Machine Learning*. Springer, Heidelberg (1997)
13. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005), <http://www.cs.waikato.ac.nz/ml/weka>

14. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
15. Shrestha, D.L., Solomatine, D.P.: Experiments with AdaBoost.RT, an improved boosting scheme for regression. *Neural Computation* 18(7), 1678–1710 (2006)
16. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles. *Machine Learning* 51, 181–207 (2003)
17. Margineantu, D.D., Dietterich, T.G.: Pruning adaptive boosting. In: *Proc. 14th International Conference on Machine Learning*, pp. 211–218. Morgan Kaufmann, San Francisco (1997)
18. Rodríguez, J.J., Maudes, J., Pardo, C., García-Osorio, C.: Disturbing neighbors ensembles for regression. In: *XIII Conference of the Spanish Association for Artificial Intelligence, CAEPIA - TTIA 2009*, pp. 369–378 (2009)

Ensemble Methods and Model Based Diagnosis Using Possible Conflicts and System Decomposition*

Carlos J. Alonso-González¹, Juan José Rodríguez², Óscar J. Prieto¹,
and Belarmino Pulido¹

¹ Intelligent Systems Group (GSI), Department of Computer Science,
E.T.S.I Informática, University of Valladolid, Valladolid, Spain

² Department of Civil Engineering, University of Burgos, Burgos, Spain

Abstract. This work presents an on-line diagnosis algorithm for dynamic systems that combines model based diagnosis and machine learning techniques. The Possible Conflicts (PCs) method is used to perform consistency based diagnosis, providing fault detection and isolation. Machine learning methods are used to induce time series classifiers, that are applied on line for fault identification. The main contribution of this work is that Possible Conflicts are used to decompose the physical system, defining the input-output structure of an ensemble of classifiers. Experimental results on a simulated pilot plant show that the ensemble created from PCs decomposition has an important potential to increase the accuracy of individual classifiers for several learning algorithms. Without PCs decomposition, the best results were for another ensemble method, Stacking. These results are improved when combining Stacking with PCs decomposition.

Keywords: System Decomposition, Ensemble Methods, Stacking, Consistency Based Diagnosis, Fault Identification.

1 Introduction

In the Artificial Intelligence field, the DX community has developed Consistency Based Diagnosis, CBD, as the major paradigm for model based diagnosis [3]. CBD performs fault detection and isolation with just models for correct behavior, but the absence of fault models knowledge is partly responsible of the low discriminative power that CBD may exhibit [6]. Usually, to solve this drawback, knowledge about fault modes is introduced. In this work, we have considered the predictive approach, which uses models of fault modes to estimate faulty behavior, as in Sherlock [5] or TRANSCEND [10]. However, adding fault modes increase complexity. For N components in a system, fault isolation must discriminate among 2^N modes (ok or fault). Fault identification must discriminate among K^N modes (for an average K behavioural modes).

To avoid this problem, we propose to use CBD without fault models, and include fault identification knowledge obtained with machine learning techniques, including ensemble methods.

* This work was supported by the Project TIN2009-11326 of the Spanish Ministry of Science and Innovation.

There are several systems that couple model based diagnosis with machine learning. In [19], from the neural network perspective, a typology of approaches is presented. In [2] a review of the compilation approach is included. Currently, three basic approaches can be found in literature, with some proposals in between: compilation, residual classification and model learning. The compilation method uses machine learning to induce classifiers that model the relations from symptoms to faults. The main objective of the compilation approach is to improve the efficiency of model-based diagnosis usually to perform on-line diagnosis [2][1]. The residual classification method uses machine learning to induce classifiers that model the relation from residuals to faults, with the objective of improving the robustness of the isolation from residuals [4][9]. The model learning method uses machine learning to induce models of the system. The objective of this approach is obtaining models of complex systems whose behavior is not well known [13][9][8].

In this work, the compilation approach is used to combine CBD with machine learning techniques, maintaining the soundness of the CBD approach. CBD is in charge of fault detection and isolation, while machine learning is used for refining CBD isolation providing a first step towards fault identification: discrimination among fault modes. The identification problem is approached as a multivariate time series classification task and time series classifiers are induced off line from simulated data. This approach has been previously tested in [1].

The aforementioned method has the inconvenient that no information is used regarding which observations are relevant to each mode fault. Actually, a single classifier was built, having as inputs all available observations and as outputs all fault modes considered. This is a naive methodology that is not appropriate for systems with a large number of observations and fault modes. A better approach is to exploit the structural knowledge used by the CBD method to define the input-output structure of the classifiers.

In this work we propose to use Possible Conflicts, PCs, [15] for both Fault Detection and Isolation, and system decomposition. In the rest of the paper, we briefly describe the structural decomposition induced by PCs and the machine learning techniques used. Then we explain how PCs decomposition is used to define the structure of an ensemble of classifiers. This ensemble can be easily integrated in the CBD cycle without affecting isolation soundness. The approach is tested in a simulated scenario and systematically evaluated.

2 Possible Conflicts for On Line CBD and System Decomposition

The computation of possible conflicts is a compilation technique which, under certain assumptions, is equivalent to on-line conflict calculation in the General Diagnostic Engine [4], GDE, or Fault Detection and Isolation using ARR's obtained through structural analysis. A detailed description of consistency based diagnosis with possible conflicts can be found in [15].

The main idea behind the *possible conflict* concept is that the set of subsystems capable to generate a conflict can be identified off-line, in a three steps process:

The first one represents the system as an hyper-graph, $H_{SD} = \{V, R\}$, where V is the set of variables of the system and $R = \{r_1, r_2, \dots, r_m\}$ is a family of sub-sets in V , where each r_k represents a constraint in the model.

The second step looks for minimal over-constrained subsystems, called *Minimal Evaluation Chains (MEC)*, $H_{ec} = \{V_{ec}, R_{ec}\}$, where $V_{ec} \subseteq V$, $R_{ec} \subseteq R$. Evaluation chains are necessary conditions for a *possible conflict* to exist. Additionally, each *MEC* identifies, by definition, a subsystem of H_{SD} .

In the third step, extra knowledge is added to assure that a *MEC*, H_{ec} , can be solved using local propagation criterion. If it is possible, a *Minimal Evaluation Model (MEM)* is defined, $H_{mem} = \{V_{mem}, R_{mem}\}$, with $V_{mem} = V_{ec}$ and $R_{mem} = \{r_{1_{k1}}, r_{2_{k2}}, \dots, r_{m_{km}}\}$. $r_{i_{ki}}$ is a causal constraint obtained assigning a causality to $r_i \in R_{ec}$. The set of relations of a *MEC* with at least one *MEM* is called a *possible conflict (PC)*.

Actually a *MEM* is a computational model with discrepancy detection capability. If there is a discrepancy between predictions from those models and current observations, the possible conflict is confirmed as a real conflict. Afterwards, diagnosis candidates are obtained from conflicts following Reiter's theory.

Additionally, *MEMs* provide a mean to decompose the original system, $H_{SD} = \{V, R\}$. Each *MEM* is uniquely related to one *MEC*. And each *MEC* identifies, by definition, a subsystem of H_{SD} . Then, the set of *MEM* induces a decomposition on H_{SD} . The decomposition is not exhaustive (some variables and relations may not be included in any subsystem) neither exclusive (some relations and variables may belong to various subsystems), but it is systematic because the algorithms that compute possible conflicts find every *MEC* and *MEM*. An important property of this decomposition is that every found subsystem is minimal in the sense that no proper subsystem has discrepancy detection capability. This assures that the decomposition induced by *MEMs* is unique.

3 Time Series Classifiers for Fault Identification

In this work, fault identification is approached as a problem of multivariate time series classification. This is adequate for those dynamic systems where the set of available observations (measurements and systems parameters and settings) is fixed and no new measurement is available to refine diagnosis candidates. We restrict ourselves to discrete and finite time series of real numbers. Several systems, like industrial continuous processes, embedded or autonomous systems, satisfy these requirements. The historical values of each observable variable may be considered as a univariate time series and the set of historical values of all the variables of interest as a multivariate time series. Therefore diagnosis of past and current faults may be achieved analyzing the past multivariate time series, particularly inducing multivariate time series classifiers.

To test the effect of system decomposition, we have selected six machine learning techniques: Decision Trees (DT), Naive Bayes (NB), Support Vector Machines (SVM) (with linear or perceptron kernel), Nearest Neighbor (k-NN) (with Dynamic Time Warping, DTW, as a dissimilarity measure), and Stacking Nearest Neighbor (Stack-k-NN).

DT, NB, SVM and k-NN are standard machine learning techniques. Stack-k-NN is a variant of the Stacking ensemble method [18], adapted to generate a multivariate time series classifier from univariate classifiers. k-NN is used to classify each univariate time series that form the multivariate one. An additional classifier, Naive Bayes in this work,

is used to generalize the output of the univariate classifiers. This configuration has been tested in [1]. Experimental results have shown that Stack-k-NN outperforms the five standard machine learning algorithms considered on various data sets.

When necessary, the machine learning algorithms have been adapted to provide class confidences instead of assigning a single class. This feature will be used to combine the classifiers outputs when decomposition selects more than one classifier.

4 Attribute and Class Selection via Possible Conflicts

In previous works, knowledge about the system to be diagnosed has not been used to decide the input-output structure of the classifiers. A single global classifier, *Classifier*, was constructed. Let n be the number of available observations. Then, the input space of the classifiers is the multivariate time series space $I = TS_1 \times TS_2 \times \dots \times TS_n$, with TS_i the time series space associated to observation i . Respecting fault modes, we assume system detectability, and only consider fault modes that may be detected by some possible conflict, according to the fault signature matrix of the system. Let's C_i be the set of fault models associated to PC_i . Then, the set of classes, C , is defined as $C = \cup_i C_i, i = 1, 2, \dots, m$, with m the number of PC s. Therefore, the classifier *Classifier*, applies I into C :

$$\text{Classifier} : I \longrightarrow C$$

Although Stack-k-NN behaves satisfactory with this simple setting for a small size problem [1], it is not to be expected that the method may scale up to systems with hundreds of observations and fault modes.

On practical applications, a great effort is usually done to analyze the characteristics of the problem. It is well known [7,9,16] that the induction of classifiers is increasingly difficult with the number of classes to consider and with the number of attributes; particularly harmful is the presence of irrelevant and/or redundant attributes. A lot of work has been done to automate this analysis, specially in the field of attribute selection. However, it is still claimed that a thorough knowledge of the problem and the help of problem experts is necessary in all but simple applications [17].

One obvious way of simplifying the problem is decomposing the physical system, looking for subsystems where the space I has lower dimension and the set C has lower cardinality. The Possible Conflict approach provides a systematic method to decompose a physical system.

Possible Conflict Decomposition: Class Decomposition. We have shown in section 2 that the set of *MEM*s induce a unique decomposition on the system. Because each MEM_i is only sensible to the mode faults of C_i , we have that $|C_i| \leq |C|$. This provides a first decomposition step that reduces the number of classes of a classifier. If we have m Possible Conflicts, then we can replace the single global classifier, *Classifier*, by m local classifiers, *Classifier- C_i* , where the suffix C indicates that decomposition has been applied to class selection and subindex i identifies the PC used. The structure of these classifiers is:

$$\text{Classifier-}C_i : I \longrightarrow C_i$$

These local classifiers, *Classifier-C_i*, still work on the same input space, but have the potential to work with a smaller number of classes.

Possible Conflict Decomposition: Attribute and Class Decomposition. The decomposition may be carried out a step further, to the level of the *MEM* subsystem. Now, the *MEMs* project the input space I into m subspaces I_i , with m the number of *MEMs* and I_i the space of time series associated to the observations of *MEM_i*, that is, the observations of V_{mem_i} . The projections are not exclusive and several subspaces may share several dimensions, that is, observations. Note also that some dimensions of I may not be projected in any subspace.

With this last level of decomposition, the original global classifier, *Classifier*, is replaced by m local classifiers, *Classifier-AC_i*, where the suffix *AC* indicates that decomposition has been applied to classes and observations (attributes). The structure of these classifiers is:

$$Classifier-AC_i : I_i \longrightarrow C_i$$

Possible Conflict Decomposition: Ensemble of Classifiers. During the identification stage, several classifiers might be invoked for the same fault. Actually, the m local classifiers may be considered as an ensemble of classifiers. We define *Ensemble-C* (resp. *Ensemble-AC*) as the ensemble of m local classifiers *Classifier-C_i* (resp. *Classifier-AC_i*). Then, a mechanism is needed to combine the class prediction of the local classifiers. We use a simple average scheme. The local classifiers assign a probability to each class of their output domain and a null probability to the remainder classes. These probabilities are averaged and provided as the output of the ensemble.

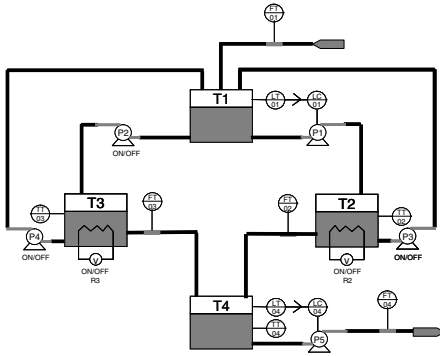
By definition, the family of C_i is a coverage of C , although not a mutually exclusive one, because it is possible to have $C_i \cap C_j \neq \emptyset$ for some $i \neq j$. Hence some degree of overlapping among the set of classes is to be expected. There is also some overlapping among the dimensions of the input space of the classifiers when attribute selection is applied. From the machine learning point of view, this is a desirable property: we may have several classifiers providing different confidence to the same classes. This has the potential to generate diversity, which is a much looked after property on the ensemble methods.

5 A Case Study

For this work, we have used the laboratory scale plant shown in figure [11](#). It is made up of four tanks $\{T_1, \dots, T_4\}$, five pumps $\{P_1, \dots, P_5\}$, and two PID controllers acting on pumps P_1, P_5 to keep the level of $\{T_1, T_4\}$ close to the specified set point. To control temperature on tanks $\{T_2, T_3\}$ we use two resistors $\{R_2, R_3\}$, respectively.

In this plant we have eleven different measurements: levels of tanks T_1 and T_4 — $\{LT01, LT04\}$ —, the control action of PID controllers on pumps $\{P_1, P_5\}$ — $\{LC01, LC04\}$ —, in-flow on tank T_1 — $\{FT01\}$ —, outflow on tanks $\{T_2, T_3, T_4\}$ — $\{FT02, FT03, FT04\}$ —, and temperatures on tanks $\{T_2, T_3, T_4\}$ — $\{TT02, TT03, TT04\}$ —. Actions on pumps $\{P_2, P_3, P_4\}$, and resistors — $\{R_2, R_3\}$ — are also known.

The plant may work with different operation modes. In this work a simple setting without recirculation —pumps $\{P_3, P_4\}$ and resistor R_2 are switch off— has been chosen.



| FM | Component | Description |
|----------|-----------|------------------------------------|
| f_1 | T_1 | Small/medium leakage in tank T_1 |
| f_2 | T_1 | Big leakage in tank T_1 |
| f_3 | T_1 | Pipe blockage T_1 (to P_1) |
| f_4 | T_1 | Pipe blockage T_1 (to P_2) |
| f_5 | T_3 | Leakage in tank T_3 |
| f_6 | T_3 | Pipe blockage T_3 (to T_4) |
| f_7 | T_2 | Leakage in tank T_2 |
| f_8 | T_2 | Pipe blockage T_2 (to T_4) |
| f_9 | T_4 | Leakage in tank T_4 |
| f_{10} | T_4 | Pipe blockage T_4 (to P_5) |
| f_{11} | P_1 | Pump failure |
| f_{12} | P_2 | Pump failure |
| f_{13} | P_5 | Pump failure |
| f_{14} | R_3 | Resistor failure in tank T_3 |

Fig. 1. Diagram of the plant and fault modes considered

Table 1. Possible conflicts found for the laboratory plant; constraints, components, and the estimated variable for each possible conflict

| | Constraints | Components | Estimate |
|--------|-----------------------------------------------|----------------------|----------|
| PC_1 | $t_{1dm}, t_{1fb1}, t_{1fb2}$ | T_1, P_1, P_2 | LT01 |
| PC_2 | $t_{1fb1}, t_{2dm}, t_{2f}$ | T_1, T_2, P_1 | FT02 |
| PC_3 | $t_{1fb1}, t_{1dE}, t_{2dE}, t_{2dm}$ | T_1, P_1, T_2 | TT02 |
| PC_4 | $t_{1fb2}, t_{3dm}, t_{3f}$ | T_1, P_2, T_3 | FT03 |
| PC_5 | $t_{1fb2}, t_{1dE}, t_{3dE}, t_{3dm}, r_{3p}$ | T_1, P_2, T_3, R_3 | TT03 |
| PC_6 | t_{4dm} | T_4 | LT04 |
| PC_7 | t_{4fb} | T_4, P_5 | FT04 |

We have used common equations in simulation for this kind of process:

1. t_{dm} : mass balance in tank t .
2. t_{dE} : energy balance in tank t .
3. t_{fb} : flow from tank t to pump.
4. t_f : flow from tank t through a pipe.
5. r_p : resistor heat transfer.

Based on these equations we have found the set of possible conflicts shown in table 1. In the table, second column shows the set of constraints used in each possible conflict, which are minimal with respect to the set of constraints. Third column shows those components involved. Fourth column indicates the estimated variable for each possible conflict. We have considered the fourteen fault modes shown in figure 1.

Class Decomposition. Possible conflicts related to fault modes are shown in the theoretical fault signature matrix shown in table 2. It should be noticed that these are the fault mode classes which can be distinguished for fault identification. In the localization stage, the following pair of faults $\{f_1, f_2\}$, $\{f_4, f_{11}\}$, $\{f_3, f_{12}\}$, and $\{f_{10}, f_{13}\}$ can not be separately isolated.

Table 2. PCs and their associated fault modes and observations

| | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 | f_7 | f_8 | f_9 | f_{10} | f_{11} | f_{12} | f_{13} | f_{14} | FT01 | FT02 | FT03 | FT04 | LT01 | LT04 | LC01 | LC04 | TT02 | TT03 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|------|------|------|------|------|------|------|------|------|------|
| PC_1 | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | | | 1 | | | | 1 | | 1 | | | |
| PC_2 | | | | 1 | | | 1 | 1 | | | 1 | | | | | 1 | | | 1 | | 1 | | | |
| PC_3 | | | | 1 | | | 1 | | | | 1 | | | | 1 | 1 | | | 1 | | 1 | | 1 | |
| PC_4 | | | 1 | | 1 | 1 | | | | | | 1 | | | | | 1 | | 1 | | | | | |
| PC_5 | | | 1 | | 1 | | | | | | | 1 | | 1 | 1 | 1 | | | | | | | | 1 |
| PC_6 | | | | | | | | | 1 | | | | | | | 1 | 1 | 1 | | 1 | | | | |
| PC_7 | | | | | | | | | | 1 | | | 1 | | | | | 1 | | 1 | | 1 | | |

The fault signature matrix provides the class decomposition. For each PC_i , C_i includes the fault modes marked 1 on the corresponding row. For instance, $C_3 = \{f_4, f_7, f_{11}\}$.

Attribute and Class Decomposition. The input an output observations of each possible conflict, obtained from each MEM , are shown in table 2. Now, each subspace I_i includes the observations marked 1 on the corresponding column. For instance, $I_3 = TS_{FT01} \times TS_{FT02} \times TS_{LT01} \times TS_{LC01} \times TS_{TT02}$.

6 Experimental Evaluation

We have resorted to a detailed, non linear quantitative simulation of the plant. We have run twenty simulations for each class, adding 2.5% noise in the sensors readings. We have modeled each fault class with a parameter in the $[0, 1]$ range. Each simulation lasted 900 seconds. We randomly generate the fault magnitude, and its origin, in the interval $[180, 300]$. The sample rate is 3 seconds. Since we just have eleven observations, then each simulation will provide eleven series of three hundred numeric elements. For complexity reasons, we only consider single faults. We also have assumed that the system is in stationary state before the fault appears.

To evaluate the viability of the Possible Conflict decomposition, a simplified scenario has been developed. The simulation data has been used to tune the possible conflicts thresholds. Hence, each instance of a fault mode simulation confirms the possible conflicts as indicated in the fault signature matrix. This has the inconvenient of introducing some bias on the evaluation method, because in a real scenario a possible conflict may not be activated when a fault mode is present. Consequently, the empirical results only compares the behavior of the different classifiers in the ideal case when fault detection and isolation is error free. Time series classifiers are invoked once real conflicts have been confirmed with a fragment of series from t to the $\min(\text{current time}, t + \text{maximum series length})$, with t a time instant previous to fault detection. When decomposition is applied, information on confirmed conflicts is also provided to the ensemble method.

All the experiments have been performed on the WEKA tool [17]. Error estimations and hypotheses testing have been obtained with the corrected resampled t-test. Instead of standard cross validation, we have made fifteen training-test iterations. In each iteration,

Table 3. Accuracies of the methods. The symbol “●” indicates that the result is significantly worse than the method with the best result for the corresponding column.

| Method | Class Attribute | | series length | | | | |
|---------------|-----------------|-----|---------------------|---------------------|---------------------|---------------------|---------------|
| | | | 30% | 40% | 50% | 70% | 100% |
| Decision Tree | No | - | ●65.95 (8.42) | ●94.76 (3.79) | ●90.95 (5.03) | ●94.29 (5.19) | ●93.81 (5.13) |
| Decision Tree | Yes | No | ●87.14 (6.01) | ●96.90 (1.84) | ●95.00 (4.00) | 96.67 (4.15) | 96.43 (3.82) |
| Decision Tree | Yes | Yes | ●88.81 (4.65) | 96.43 (3.02) | 96.67 (3.16) | ●95.71 (3.87) | ●95.48 (4.15) |
| Naive Bayes | No | - | ●57.86 (4.71) | ●88.10 (4.99) | ●91.90 (4.77) | ●92.38 (4.84) | ●82.86 (5.43) |
| Naive Bayes | Yes | No | ●87.38 (4.24) | ●95.71 (3.62) | 97.14 (2.77) | 97.38 (2.85) | 97.86 (2.63) |
| Naive Bayes | Yes | Yes | ●89.05 (3.16) | 98.10 (2.29) | 97.62 (2.20) | 97.14 (2.41) | ●97.14 (2.77) |
| SVM linear | No | - | ●42.62 (5.48) | ●83.10 (4.15) | ●87.14 (4.44) | ●89.52 (3.69) | ●94.05 (3.97) |
| SVM linear | Yes | No | ●74.52 (7.25) | ●93.33 (3.27) | 97.14 (2.41) | 98.81 (1.74) | ●99.76 (0.92) |
| SVM linear | Yes | Yes | ●76.90 (6.17) | ●89.52 (3.43) | ●90.24 (3.69) | ●90.24 (3.43) | ●90.71 (3.77) |
| SVM perc. | No | - | ●50.24 (7.33) | ●82.62 (4.24) | ●86.90 (4.61) | ●88.57 (4.52) | ●88.81 (4.24) |
| SVM perc. | Yes | No | ●88.10 (4.20) | 98.10 (1.84) | 99.05 (1.63) | 99.29 (1.48) | 99.29 (1.48) |
| SVM perc. | Yes | Yes | ●89.29 (3.02) | 97.38 (2.51) | 98.57 (1.81) | 98.57 (1.81) | 98.33 (1.84) |
| 1-NN | No | - | ●48.81 (7.72) | ●84.52 (7.35) | ●85.00 (7.17) | ●86.67 (6.54) | ●87.62 (6.31) |
| 1-NN | Yes | No | ●86.43 (6.91) | ●93.57 (3.87) | ●93.33 (4.65) | ●92.62 (4.57) | ●94.05 (4.20) |
| 1-NN | Yes | Yes | 96.43 (4.27) | 99.52 (1.26) | 98.81 (1.74) | 98.81 (1.74) | 98.10 (1.84) |
| Stack-1-NN | No | - | ●63.81 (7.50) | ●96.67 (2.51) | 97.86 (2.26) | 98.10 (2.65) | 99.05 (1.63) |
| Stack-1-NN | Yes | No | ●89.76 (3.79) | 97.62 (2.58) | 98.33 (1.84) | 98.81 (2.20) | 99.52 (1.26) |
| Stack-1-NN | Yes | Yes | 94.29 (3.52) | 99.29 (1.48) | 99.29 (1.48) | 99.52 (1.26) | 99.52 (1.26) |

we have randomly selected 90% of available data for the training set and 10% for the test set. This experimental setting is proposed in [12].

We have tested the system for 30, 40, 50, 70, and 100% of the total time series length. With 30% of the series the system is beginning the transition to a new stationary state, that is nearly reached with a 70%. Table 3 shows the result obtained. The column *Class* (resp. *Attribute*) indicates if Class decomposition has been applied (resp. Attribute decomposition).

Without decomposition, first row of each learning method, DT and Stack-1-NN provide the best results. Both methods are less sensitive to irrelevant attributes, specially Stack-1-NN.

Ensembles based on class decomposition systematically increase the accuracy of the classifiers. Improvement is particularly important for series length of 30% for all base classifiers, when fault effects are starting to manifest. Enhancement is also notable for SVM with kernel perceptron as base classifier, which achieves the highest accuracy except for lengths of 30% and 100%, where Stack-1-NN and SVM with liner kernel, respectively, obtains slightly better accuracies.

The behavior of ensembles based on class and attribute decomposition depends on the base classifiers and the considered length of the series. Accuracies always increase for 30% series length. This is an interesting result, because it facilitates precise early fault isolation. For other lengths, DT and NB ensembles may perform slightly worse. Accuracy of SVM with kernel perceptron slightly decreases, while SVM with linear kernel clearly worsen. Stack-1-NN still improves. The base classifier that benefits most of attribute decomposition is 1-NN. Somehow this is something to be expected, because nearest neighbor algorithms degrade severely with irrelevant attributes.

Discussion. Several ensemble methods have been tested in this work. Experimental results indicate that in the absence of structural knowledge, Stacking-1-NN is a good choice. As it was expected, ensembles based on structural knowledge improve on global

base classifiers. Ensembles based on class decomposition with SVM kernel perceptron as base classifier provides excellent results, only exceed by ensembles of Staking 1-NN for 30% series length.

Ensembles based on class and attribute decomposition provides the maximum accuracy, except for 100% series length, where SVM with linear kernel wins by a small margin. Table 3 shows in **bold** the methods with highest accuracy for each series length. Significance tests with 0.05 level have been performed against all the other methods in the corresponding column. For 30, 40%, 1-NN is the preferred method: accurate and simple. For other series length, ensembles of Stacking 1-NN seems slightly better.

And additional benefit of attribute decomposition, not considered in this evaluation, is that it may reduce by an important factor the training time (classification time if 1-NN) respect to class decomposition, because all the local classifiers works on a smaller input space if attribute decomposition is applied.

7 Conclusions and Further Work

This work has shown how Possible Conflicts technique may be used to decompose a system. The decomposition defines the structure of an ensemble of classifiers that may be integrated on line in the Consistency Based Diagnosis cycle, providing fault identification information. The integration relies on the compilation approach of Machine Learning and Model Based Diagnosis.

The proposal has been evaluated on a simplified scenario on various machine learning methods. Experimental results show that Class decomposition improves the accuracy of the classifiers. The effect of Attribute decomposition is not so consistent, although improves the accuracy of the classifiers at the first stages of the diagnosis processes.

Future work requires a thorough evaluation of the proposal on the same pilot plant used in this paper. Data from real faults have been collected. Also, additional simulation data must be generated, to evaluate the behavior of the classifiers independently of the possible conflicts detection and isolation accuracy.

References

1. Alonso, C.J., Prieto, O.J., Rodríguez, J.J., Bregón, A., Pulido, B.: Stacking dynamic time warping for the diagnosis of dynamic systems. In: Borrajo, D., Castillo, L., Corchado, J.M. (eds.) CAEPIA 2007. LNCS (LNAI), vol. 4788, pp. 11–20. Springer, Heidelberg (2007)
2. Console, L., Picardi, C., Dupre, D.T.: Temporal decision trees: Model-based diagnosis of dynamic systems on-board. *Journal of Artificial Intelligence Research* 19, 469–512 (2003)
3. de Kleer, J., Mackworth, A.K., Reiter, R.: Characterising diagnosis and systems. In: *Readings in Model Based Diagnosis*, pp. 54–65. Morgan Kaufmann, San Francisco (1992)
4. de Kleer, J., Williams, B.C.: Diagnosing multiple faults. *Artificial Intelligence* 32, 97–130 (1987)
5. de Kleer, J., Williams, B.C.: Diagnosing with behavioral modes. In: *Eleventh International Joint Conference on Artificial Intelligence, IJCAI 1989* (1989)
6. Dressler, O., Struss, P.: The consistency-based approach to automated diagnosis of devices. In: *Principles of Knowledge Representation*, pp. 269–314. CSLI Publications, Stanford (1996)

7. Frank, E., Kramer, S.: Ensembles of nested dichotomies for multi-class problems. In: Proceedings of the 21st International Conference on Machine Learning (2004)
8. Lerner, U., Parr, R., Koller, D., Biswas, G.: Bayesian fault detection and diagnosis in dynamic systems. In: Proceedings of the AAAI/IAAI, pp. 531–537 (2000)
9. Li, T., Zhang, C., Ogihara, M.: A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *BIOINFORMATICS* 20(15), 2429–2437 (2005)
10. Mosterman, P., Biswas, G.: Diagnosis of continuous valued systems in transient operating regions. *IEEE T. Syst. Man. Cy. B.* 29(6), 554–565 (1999)
11. Murphey, Y.L., Masrur, M.A., Chen, Z., Zhang, B.: Model-based fault diagnosis in electric drives using machine learning. *IEEE/ASME Transactions On Mechatronics* 11(3), 290–303 (2006)
12. Nadeau, C., Bengio, Y.: Inference for the generalization error. *Machine Learning* 52(3), 239–281 (2003)
13. Patton, R.J., Chen, J., Siew, T.M.: Fault diagnosis in nonlinear dynamic systems via neural networks. In: Proc. IEE Int. Conf. Control 1994, vol. 2, pp. 1346–1351 (1994)
14. Pernestål, A., Nyberg, M., Wahlberg, B.: A bayesian approach to fault isolation with application to diesel engine diagnosis. In: Proc. of the 17th International Workshop on Principles of Diagnosis, DX 2006, pp. 211–218 (2006)
15. Pulido, B., Alonso González, C.: Possible conflicts: a compilation technique for consistency-based diagnosis. *IEEE T. Syst. Man Cy. B* 34(5), 2192–2206 (2004)
16. Rokach, L.: Decomposition methodology for classification tasks: a meta decomposer framework. *Pattern Anal. Applic.* 9, 257–271 (2006)
17. Witten, I., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)
18. Wolpert, D.H.: Stacked generalization. *Neural Networks* 5(2), 241–260 (1992)
19. Yu, D.L., Gomm, J.B., Williams, D.: Sensor fault diagnosis in a chemical process via RBF neural networks. *Control Engineering Practice* 7, 49–55 (1999)

Entropy-Based Evaluation Relaxation Strategy for Bayesian Optimization Algorithm

Hoang Ngoc Luong, Hai Thanh Thi Nguyen, and Chang Wook Ahn

School of Information and Communication Engineering, Sungkyunkwan University
300 Cheoncheon-dong, Suwon 440-746, Republic of Korea
{ngochoang, haintt86, cwan}@skku.edu

Abstract. Bayesian Optimization Algorithm (BOA) belongs to the advanced evolutionary algorithms (EA) capable of solving problems with multivariate interactions. However, to attain wide applicability in real-world optimization, BOA needs to be coupled with various efficiency enhancement techniques. A BOA incorporated with a novel entropy-based evaluation relaxation method (eBOA) is developed in this regard. Composed of an on-demand evaluation strategy (ODES) and a sporadic evaluation method, eBOA significantly reduces the number of (fitness) evaluations without imposing any larger population-sizing requirement. Experiments adduce the grounds for its significant improvement in the number of evaluations until reliable convergence. Furthermore, the evaluation relaxation does not negatively affect the scalability performance.

Keywords: Estimation of Distribution Algorithms, Bayesian Optimization Algorithm, Evaluation Relaxation, On-Demand Evaluation.

1 Introduction

Being an outgrowth from classical genetic algorithms, Estimation of Distribution Algorithms (EDAs) [1] have emerged as a robust optimization methodology. Instead of obtaining new solutions by traditional recombination and mutation operators, EDAs gradually develop and maintain a probability distribution of promising candidates. New offspring can be generated by sampling this underlying distribution. Bayesian Optimization Algorithm (BOA) [2] belongs to the class of multivariate EDAs. BOA is capable of tackling multivariate interaction problems decomposable into subproblems of bounded difficulty. Extended Compact Genetic Algorithm (ECGA) [3], Factorized Distribution Algorithm (FDA) [4], and Estimation of Bayesian Network Algorithm (EBNA) [4] are other examples of state-of-the-art EDAs. The strength of BOA is brought into the continuous world by Real-coded Bayesian Optimization Algorithm (rBOA) [5][6].

A standard BOA is composed of the following steps.

Step 1: Set $i \leftarrow 0$. Randomly generate first population $\mathcal{P}(0)$. Evaluate $\mathcal{P}(0)$.

Step 2: Select a parent set $\mathcal{S}(i)$ of promising solutions from $\mathcal{P}(i)$.

Step 3: Learn a Bayesian network $\mathcal{B}(i)$ from $\mathcal{S}(i)$.

- Step 4:** Generate new solutions population $\mathcal{O}(i)$ by sampling from $\mathcal{B}(i)$.
Step 5: Evaluate entire $\mathcal{O}(i)$.
Step 6: Create $\mathcal{P}(i + 1)$ by replacing some solutions of $\mathcal{P}(i)$ with $\mathcal{O}(i)$.
Step 7: Set $i \leftarrow i + 1$. If the termination criteria are not met, go to **Step 2**.

BOA uses Bayesian networks [7] to model the regularities of the promising solutions. The process of learning a Bayesian network from a selected population is concisely presented in Pelikan et al. [2]. A Bayesian network encodes the following joint probability distribution

$$p(X_0, X_1, \dots, X_{n-1}) = \prod_{i=0}^{n-1} p(X_i | \Pi_i), \quad (1)$$

where each node X_i is the i^{th} random variable, Π_i is the set of parent nodes of X_i in the network (from each node of the set Π_i , there exists an edge directing into X_i), and $p(X_i | \Pi_i)$ is the conditional probability of X_i given its parents Π_i . Each variable X_i has a corresponding conditional probability table (CPT) storing its conditional probabilities concerning all possible values of Π_i .

Despite being a robust global black box optimizer, BOA has two potential bottlenecks involved in its process, the Bayesian networks construction and the fitness evaluations. In real-world application, the latter one costs a considerable amount of time and resources. Thus, various efficiency enhancement techniques for EDAs have been developed to reduce the number of (fitness) evaluations required for discovering the global optimum [8,9,10]. In this paper, a novel fitness evaluation relaxation strategy for BOA is proposed. To this end, the concept of the entropy measurement of (sub)populations is utilized.

The paper is organized as follows. Section 2 briefly describes related works covering various evaluation relaxation approaches for BOA. The concept of the entropy of a certain population and its inherent characteristics are discussed in Sect. 3. Our evaluation relaxation strategy is presented in Sect. 4. Experiments and results are shown in Sect. 5. Section 6 concludes the paper and highlights the future work.

2 Related Work

In BOA, the conditional dependencies between random variables are encoded in a directed acyclic graph. Taking advantage of such high level of abstraction, various techniques have been proposed to achieve evaluation relaxation for BOA.

Fitness inheritance in BOA [9] builds surrogate fitness models based on Bayesian networks. At each iteration, only a certain proportion of newly generated solutions are determined using the actual evaluation function. Fitness values of the remaining offspring are estimated by the partial contributions of each variable $X_i = x_i$ with respect to its parents $\Pi_i = \pi_i$ as follows

$$f_{\text{estimation}}(X_0, X_1, \dots, X_{n-1}) = \bar{f} + \sum_{i=0}^{n-1} (\bar{f}(X_i | \Pi_i) - \bar{f}(\Pi_i)), \quad (2)$$

where \bar{f} denotes the average value of all individuals used to construct the fitness model, $\bar{f}(X_i|\Pi_i)$ is the average fitness of solutions having a certain configuration of $X_i = x_i$ and $\Pi_i = \pi_i$, and $\bar{f}(\Pi_i)$ is the average fitness of solutions with a particular instance of $\Pi_i = \pi_i$.

While being proved to be a good interpolation for test problems, the construction of the above surrogate fitness model results in a greater population-sizing requirement. Furthermore, the algorithm's performance when enlarging the problem size has not been rigorously tested in the research [9].

BOA with substructural hillclimbing [10] utilizes the above-mentioned surrogate fitness model to estimate the fitness of a mutated individual at each step of the local search until it reaches a local optimum. While a reduction in the number of actual evaluations is achieved, this hillclimbing also requires larger population sizes. Besides, when the problem size is increased (when $l > 80$ as reported in [10]), the speed-up of the algorithm slows down.

In this paper, we measure the entropy value of the (sub)population at each iteration of BOA. Based on this measurement, a novel efficiency enhancement strategy is proposed to accelerate the BOA in terms of the number of evaluations.

3 Entropy Measurement of Populations

Based on the dependencies encoded in a Bayesian network, the entropy $H(\mathbf{X})$ of a certain population of BOA is derived as follows:

$$\begin{aligned}
 H(X_0, X_1, \dots, X_{n-1}) &= \sum_{i=0}^{n-1} H(X_i|\Pi_i) = \sum_{i=0}^{n-1} \sum_{\pi_i \in \mathcal{Q}_i} p(\pi_i) H(X_i|\Pi_i = \pi_i) \\
 &= - \sum_{i=0}^{n-1} \sum_{\pi_i \in \mathcal{Q}_i} p(\pi_i) \sum_{x_i \in \mathcal{X}_i} p(x_i|\pi_i) \log_2 p(x_i|\pi_i) \\
 &= - \sum_{i=0}^{n-1} \sum_{\pi_i \in \mathcal{Q}_i} \sum_{x_i \in \mathcal{X}_i} p(x_i, \pi_i) \log_2 p(x_i|\pi_i) \\
 &= - \sum_{i=0}^{n-1} \sum_{\pi_i \in \mathcal{Q}_i} \sum_{x_i \in \mathcal{X}_i} \left(\frac{m(x_i, \pi_i)}{N} \right) \log_2 \left(\frac{m(x_i, \pi_i)}{m(\pi_i)} \right), \quad (3)
 \end{aligned}$$

where \mathcal{Q}_i is the set of all possible instances of Π_i (parent nodes of X_i), \mathcal{X}_i denotes the set of all possible values of X_i , N is the population size, $m(x_i, \pi_i)$ is the number of individuals having (X_i, Π_i) set to (x_i, π_i) , and $m(\pi_i)$ is the number of individuals having Π_i equal to π_i . Research shows that such an entropy measurement can be used to determine convergence criteria for BOA [11].

Using (3), we can compute the entropy of some particular portions of the population with respect to the current network. Figure 1 shows that the entropy of the better half of the population (parents set) decreases after every iteration. When the BOA converges, this entropy value reaches its minimum value 0. This result is naturally logical as the BOA continuously narrows its sampling towards

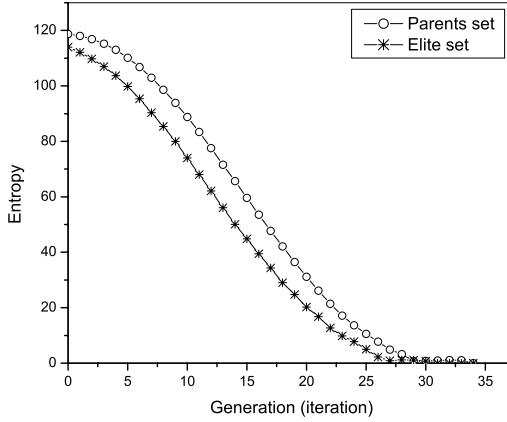


Fig. 1. Entropy reduction in a BOA solving 120-bit trap-5 problem

a certain promising region of the search space. The closer the algorithm moves to a specific convergence point, the more predictable it becomes. Thus, the inherent randomness of the probability distribution in the population gradually decreases.

We define an elite set as a set of the most promising individuals selected from the population. In Fig. 1, the top $\tau = 5\%$ of the population (in terms of fitness value) are selected as the elite set. This set also exhibits a tendency to decrease its entropy measurement after every iteration. Obviously, this regional entropy is always smaller than the overall entropy of the entire population, and it reaches the minimum value 0 sooner. From such observation, we conjecture that a promising candidate solution is an individual whose appearance in the elite set causes a reduction in the entropy measurement of that set. The next section describes how this conjecture is used in our evaluation relaxation strategy.

4 Entropy-Based Evaluation Relaxation Strategy for Bayesian Optimization Algorithm (eBOA)

4.1 The Algorithm

The proposed eBOA is described as follows:

Step 1: Initialization: Set $i \leftarrow 0, t \leftarrow 0$.

Randomly generate the initial population $\mathcal{P}(0)$. Evaluate $\mathcal{P}(0)$.

Step 2: Selection: Select a set $\mathcal{S}(i)$ of promising solutions from $\mathcal{P}(i)$.

Create an elite set $\mathcal{E}(i)$ of solutions from $\tau\%$ of $\mathcal{P}(i)$ having the highest actual fitness values.

Step 3: Model construction: Construct a Bayesian network $\mathcal{B}(i)$ fit for $\mathcal{S}(i)$.

Based on $\mathcal{B}(i)$, compute the entropy $\mathcal{H}(i)$ of $\mathcal{E}(i)$ by (3).

Step 4: Create a set of offspring $\mathcal{O}(i)$ by sampling $\mathcal{B}(i)$.

Step 5: Perform on-demand evaluation for $\mathcal{O}(i)$.

Step 6: Replace some solutions of $\mathcal{P}(i)$ by $\mathcal{O}(i)$ to create $\mathcal{P}(i+1)$.

Step 7: Set $i \leftarrow i+1$. If the termination criteria are not met, go to **Step 2**.

The on-demand evaluation strategy used in **Step 5** is defined as follows. Let t be a counter variable, and κ be the interval of sporadic evaluations.

Case 1: If $\mathcal{H}(i) \leq \frac{\mathcal{H}(0)}{2}$ and $t < \kappa$,

Consider each newly generated offspring X of $\mathcal{O}(i)$,

1. Put X into $\mathcal{E}(i)$ to create $\mathcal{E}'(i)$. Compute the entropy $\mathcal{H}'(i)$ for $\mathcal{E}'(i)$.
2. If $\mathcal{H}'(i) \leq \mathcal{H}(i)$, estimate

$$f_{\text{estimation}}(X) = f(Y), \quad Y \in \mathcal{E}(i), \forall Z \in \mathcal{E}(i), f(Y) \leq f(Z) . \quad (4)$$

Otherwise, evaluate $f(X)$.

3. $t \leftarrow t+1$.

Case 2: If $\mathcal{H}(i) > \frac{\mathcal{H}(0)}{2}$ or $t = \kappa$,

1. Evaluate all offspring of $\mathcal{O}(i)$.
2. $t \leftarrow 0$.

Next, we explain the two main ideas of our entropy-based evaluation relaxation, *the on-demand evaluation strategy* and *the sporadic evaluation*.

4.2 On-Demand Evaluation Strategy (ODES)

Evaluating the fitness of a candidate solution is an expensive operation, thus we should only do so when necessary. As stated in Sect. 3, an individual is a promising candidate solution if its appearance achieves a reduction in the entropy value of the elite set. If an individual causes such an entropy reduction, it has the same characteristics (in terms of building blocks) with other candidate solutions in the elite set. Thus, it can be selected without being evaluated by the actual fitness function. Otherwise, the individual does not belong to the elite set, and it should be evaluated to obtain the correct fitness value.

If an individual is deemed to belong to the elite set, it should be assigned a high fitness value. We have performed various fitness assignment methods for a selected individual, such as assigning the fitness value of the closest individual (in Hamming distance) or using the median value of the elite set. However, their empirical results (which are not shown here) exhibit no significant difference. In this work, we choose to assign the fitness value of the worst candidate solution in the elite set to that individual. The justification for doing so is to minimize the severity of incorrect estimation errors. For the same reason, the elite set should only be selected from the candidates evaluated by the actual fitness function.

The condition $\mathcal{H}(i) \leq \frac{\mathcal{H}(0)}{2}$ denotes that we wait until the entropy of the elite set decreases by half of its original value before applying ODES. Before this juncture, the randomness (or unpredictability) of the elite set still remains high. If ODES is applied earlier, the estimation power is not strong enough to give good approximations. This would result in slower convergence and more

actual evaluations. On the other hand, if we delay the application of ODES until later iterations, the algorithm almost converges and we cannot achieve the maximal reduction in the number of fitness evaluations. Thus, the iteration i when $\mathcal{H}(i) \leq \frac{\mathcal{H}(0)}{2}$ is a rational choice. Experiments supporting this choice are given in Sect. 5.

4.3 Sporadic Evaluation

The more our on-demand evaluation strategy is continuously used, the more estimation errors are accumulated. Even though our model improves its accuracy in later iterations, the appearances of incorrectly estimated individuals prevent the algorithm from convergence. One simple method to eliminate such accumulated errors is to sporadically perform complete evaluations of entire offspring populations. Instead of evaluating the whole population at each iteration as in the standard BOA, this should be performed only after every some generations have passed.

Methods with more complicated calculations can be developed to determine appropriate iterations to apply the sporadic evaluation. In this paper, however, we simply choose the criterion that after every κ iterations, the actual fitness function should be used to evaluate all the newly generated offspring. The condition $t < \kappa$ in the above-mentioned algorithm denotes this criterion.

5 Experiments and Discussion

5.1 Test Problems

In this work, the OneMax function and the Concatenated 5-bit trap function are taken as the test problems. OneMax defines the fitness value of an individual as simply its sum of all the bits:

$$f_{\text{onemax}}(X_0, X_1, \dots, X_{n-1}) = \sum_{i=0}^{n-1} X_i, \quad (5)$$

where $(X_0, X_1, \dots, X_{n-1})$ is an input binary string of n bits. The optimal solution for an n -bit OneMax problem is a binary string containing all 1s. Since OneMax is an easy problem for evolutionary algorithms, both BOA and eBOA must be able to work well on this problem.

A concatenated 5-bit trap is composed of several trap functions of order 5. Each trap-5 function is defined as follows:

$$f_{\text{trap5}}(u) = \begin{cases} 5 & \text{if } u = 5 \\ 4 - u & \text{if } u < 5 \end{cases}, \quad (6)$$

where u is the number of bits having value 1 in a trap-5 function. The values of all the traps are added together to form the overall fitness value. An n -bit trap-5 function has one global optimum (a string of all 1s) and $(2^{n/5} - 1)$ local optima.

The difficulty in optimizing this function is that in each 5-bit trap function, all 5 bits have to be considered together. A robust EDA has to be able to discover the structure of the problem because all statistics of any lower orders lead the algorithm away from the global optimum.

5.2 Experimental Results

Both the OneMax and the Concatenated 5-bit trap problems are employed to test the standard BOA and eBOA. The problem sizes are enlarged from 30, to 60, 90, 120, and 150 bits. For each test problem and each problem size above, 30 independent experiments are performed. For each experiment, the bisection method is used to determine the minimal population size for the algorithm to obtain the optimal solution (with 100% correct bits). The convergence criterion is when the proportion of a certain value on each position reaches 99%. The truncation selection with $\tau = 50\%$ is used to select the better half of population as the parents set. The offspring replace the worse half of the old population. In all runs of eBOA, we select the elite set with the top $\tau = 5\%$ from the old population, and we perform sporadic evaluation after every $\kappa = 5$ iterations.

Figure 2 and Table 1 compare the performance of BOA and eBOA on the OneMax problem. On average, eBOA requires 85% of the number of fitness evaluations of BOA until convergence. Although in some test cases, the average number of evaluations of both algorithms are not significantly different, eBOA is still proved to be competitive with BOA in solving the OneMax problem.

Figure 3 and Table 2 compare the performance of standard BOA and eBOA on the trap-5 problem. The results prove that our eBOA achieves a significant reduction in the number of evaluations until convergence. On average, eBOA only needs 76% of the number of fitness evaluations of BOA. Furthermore, eBOA does

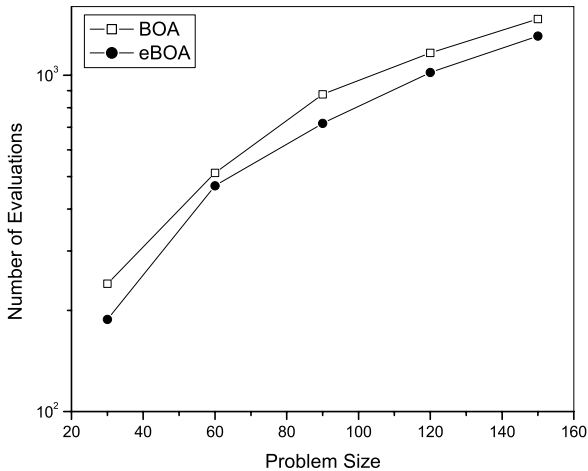
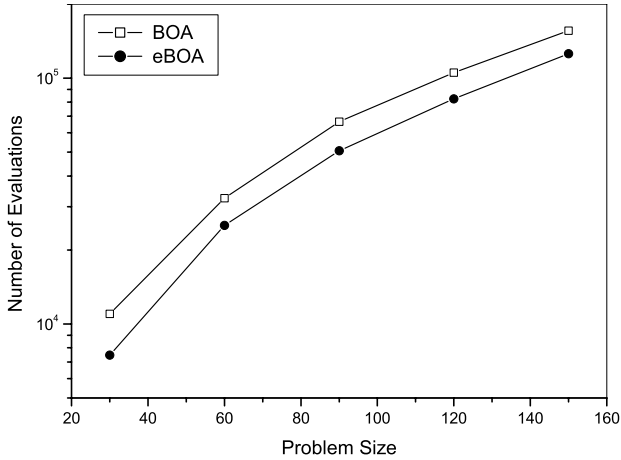


Fig. 2. Performance of BOA and eBOA on OneMax problem

Table 1. Statistical comparison of algorithms for OneMax problem

| Size | 30 | 60 | 90 | 120 | 150 |
|------------|-------------------------------------|---------|--------------------------|-------------------|-------------------|
| BOA | 240.6 | 512.4 | 877.0 | 1166.4 | 1470.4 |
| σ | (55.6) | (75.2) | (191.7) | (204.7) | (235.2) |
| eBOA | 188.6 | 469.0 | 719.2 | 1019.9 | 1308.9 |
| σ | (45.6) | (82.8) | (95.3) | (113.7) | (185.1) |
| p -value | Statistical t -test: (BOA - eBOA) | | | | |
| | $7.32\text{E-}4^\dagger$ | 0.04619 | $8.55\text{E-}4^\dagger$ | 0.00431^\dagger | $1.02\text{E-}02$ |

† Significance by a paired, two-tailed test at $\alpha = 0.01$.

**Fig. 3.** Performance of BOA and eBOA on trap-5 problem**Table 2.** Statistical comparison of algorithms for trap-5 problem

| Size | 30 | 60 | 90 | 120 | 150 |
|------------|-------------------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| BOA | 11005.3 | 32496.4 | 66453.1 | 105245.1 | 155950.3 |
| σ | (1519.4) | (3771.1) | (5118.9) | (13198.5) | (15729.4) |
| eBOA | 7465.6 | 25156.7 | 50683.8 | 82321.7 | 125699.6 |
| σ | (1167.5) | (3984.8) | (5477.1) | (6297.1) | (14035.6) |
| p -value | Statistical t -test: (BOA - eBOA) | | | | |
| | $1.39\text{E-}11^\dagger$ | $5.35\text{E-}07^\dagger$ | $1.17\text{E-}12^\dagger$ | $1.67\text{E-}09^\dagger$ | $1.32\text{E-}08^\dagger$ |

† Significance by a paired, two-tailed test at $\alpha = 0.01$.

not compromise on scalability as the problem size increases. This experiment clearly supports the claim that eBOA outperforms the standard BOA.

Table 3 shows another interesting result obtained by eBOA in solving the trap-5 problem. While being able to reduce the number of evaluations, eBOA does not introduce any larger population-sizing requirements. Note that statistical tests demonstrate no significant difference between the population sizes of BOA and eBOA.

Table 3. Statistical comparison of population sizes for solving trap-5 problem

| Size | 30 | 60 | 90 | 120 | 150 |
|------------|-------------------------------------|---------|---------|----------|----------|
| BOA | 999.1 | 2392.8 | 4404.6 | 6257.5 | 8489.6 |
| σ | (163.4) | (278.9) | (461.1) | (1020.2) | (1016.4) |
| eBOA | 992.3 | 2632.4 | 4397.1 | 6146.5 | 8632.2 |
| σ | (152.0) | (530.7) | (567.4) | (413.3) | (1160.8) |
| p -value | Statistical t -test: (BOA - eBOA) | | | | |
| | 0.85544 | 0.05682 | 0.95358 | 0.57186 | 0.60253 |

[†] Significance by a paired, two-tailed test at $\alpha = 0.01$.

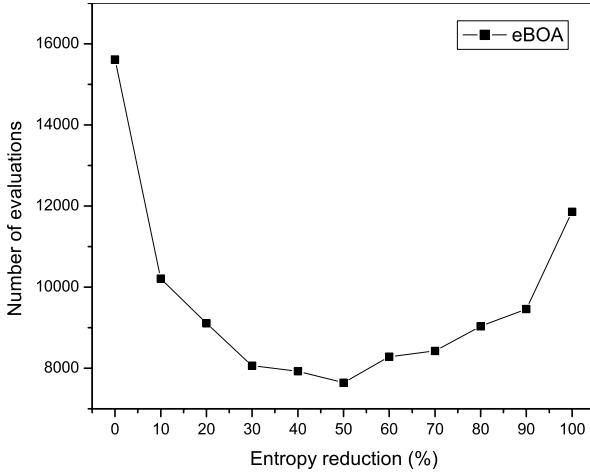


Fig. 4. Performance of eBOA with different starting points of ODES

Figure 4 provides experimental results to support our choice of the starting point to apply ODES as stated in Sect. 4.2. We perform experiments for eBOA solving the 30-bit trap-5 problem with different starting points. In terms of entropy reduction, 0% means to start ODES from the beginning of the optimization process, 100% indicates the standard BOA, and 50% comes under eBOA. When the entropy of the elite set decreases as a half of its original value, starting ODES achieves the minimal number of evaluations.

6 Conclusion

In this paper, we have presented eBOA with an entropy-based evaluation relaxation strategy. As we have observed in Sect. 3, the entropy value of the elite set gradually decreases as the algorithm moves towards the convergence point. Taking advantage of this inherent characteristic of the entropy value, we have proposed a recognition method to decide whether an individual needs to be evaluated by the actual fitness function or not. Experimental results have proved that

eBOA achieves a significant reduction in the number of fitness evaluations in comparison with the standard BOA. Moreover, it does not impose any larger population-sizing requirements. With a similar population size, the on-demand evaluation strategy can considerably accelerate the optimization process

In future work, we will investigate the effects of incorporating other estimation models into eBOA. Besides, we would like to bring the strength of on-demand evaluation strategy to other evolutionary algorithms. Such an entropy-based efficiency enhancement technique can contribute to a new line of research for EDAs in terms of evaluation relaxation.

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2009-0066229). Dr. Ahn is the corresponding author.

References

1. Pelikan, M., Goldberg, D.E., Lobo, F.G.: A Survey of Optimization by Building and Using Probabilistic Models. In: *Computational Optimization and Applications*, vol. 21, pp. 5–20. Kluwer Academic Publishers, The Netherlands (2002)
2. Pelikan, M., Goldberg, D.E., Cantu-Paz, E.: BOA: The Bayesian optimization algorithm. In: *Proceedings of GECCO 1999*, pp. 525–532. Morgan Kaufmann Publishers, San Francisco (1999)
3. Harik, G.: Linkage learning via probabilistic modeling in the ECGA. Technical Report No. 99010. IlliGAL (1999)
4. Larrañaga, P., Lozano, J.A.: *Estimation of distribution algorithms: A new tool for evolutionary computation*. Kluwer Academic Publishers, Boston (2002)
5. Ahn, C.W., Goldberg, D.E., Ramakrishna, R.S.: Real-coded Bayesian Optimization Algorithm: Bringing the Strength of BOA into the Continuous World. In: Deb, K., et al. (eds.) *GECCO 2004*. LNCS, vol. 3102, pp. 840–851. Springer, Heidelberg (2004)
6. Ahn, C.W., Ramakrishna, R.S.: On the Scalability of Real-Coded Bayesian Optimization Algorithm. *IEEE Transactions on Evolutionary Computation* 12(3), 307–322 (2008)
7. Pearl, J.: *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, San Mateo (1988)
8. Sastry, K., Lima, C.F., Goldberg, D.E.: Evaluation Relaxation Using Substructural Information and Linear Estimation. In: *Proceedings of GECCO 2006*, pp. 419–426. ACM Press, New York (2006)
9. Pelikan, M., Sastry, K.: Fitness Inheritance in Bayesian Optimization Algorithm. In: Deb, K., et al. (eds.) *GECCO 2004*. LNCS, vol. 3103, pp. 48–59. Springer, Heidelberg (2004)
10. Lima, C.F., Pelikan, M., Sastry, K., Butz, M., Goldberg, D.E., Lobo, F.G.: Substructural Neighborhoods for Local Search in the Bayesian Optimization Algorithm. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) *PPSN 2006*. LNCS, vol. 4193, pp. 232–241. Springer, Heidelberg (2006)
11. Ocenasek, J.: Entropy-Based Convergence Measurement in Discrete Estimation of Distribution Algorithms. *Studies in Fuzziness and Soft Computing*, vol. 192, pp. 39–50. Springer, Heidelberg (2006)

A New Artificial Immune System for Solving the Maximum Satisfiability Problem

Abdesslem Layeb¹, Abdel Hakim Deneche¹, and Souham Meshoul²

¹ Computer Science Department - Mentouri University Constantine, Algeria

² Information Technology Department, CCIS - King Saud University Riyadh,
Kingdom of Saudi Arabia

{adeneche, Layeb.univ}@gmail.com, meshoul@ccis.edu.sa

Abstract. In this paper we investigate the use of Artificial Immune Systems' principles to cope with the satisfiability problem. We describe ClonSAT, a new iterative approach for solving the well known Maximum Satisfiability (Max-SAT) problem. This latter has been shown to be NP-hard if the number of variables per clause is greater than 3. The underlying idea is to harness the optimization capabilities of artificial clonal selection algorithm to achieve good quality solutions for MaxSAT problem. To foster the process, a local search has been used. The obtained results are very encouraging and show the feasibility and effectiveness of the proposed hybrid approach.

Keywords: Maximum Satisfiability problem, Artificial Immune System, WalkSat.

1 Introduction

“SAT” problem is the shorthand of Boolean satisfiability problem. It is defined as the task of determining the satisfiability of a given Boolean formula by looking for the variable assignment that makes this formula evaluating to true. The Maximum Satisfiability (Max-SAT) problem is a variant of SAT problem that aims to find the variable assignment maximizing the number of satisfied clauses. In 1971, Stephen Cook [1] had demonstrated that the Max-SAT problem is NP-complete. This complex problem has several applications in different areas such as model checking, graph colouring and task planning to cite just few.

Modern Max-SAT solvers have deeply improved the techniques and algorithms to find optimal solutions. In practice there are two broad classes of algorithms for solving instances of SAT: Complete and Incomplete methods. Complete algorithms are able to verify the satisfiability or unsatisfiability of the SAT problem. They usually have an exponential complexity [2]. The most popular algorithms of this class are based on the Davis-Putnam-Loveland algorithm (DPLL) [3]; for example, a branch and bound algorithm based on DPLL is one of the most competitive exact algorithms for Max-SAT [4]. On the other hand, incomplete methods are principally based on local search and evolutionary algorithms. Incomplete methods find good quality solutions in reasonable time. Therefore, they don't guarantee optimality. This class of methods encompasses Evolutionary Algorithms (EA) [5], Stochastic Local Search (SLS) methods [6] and hybrid methods [7].

Evolutionary computation has been proven to be an effective way to solve complex engineering problems. It presents many interesting features such as adaptation, emergence and learning. Artificial neural networks, genetic algorithms and artificial immune systems are examples of bio-inspired systems used to this end. An Artificial Immune System (AIS) [8] is a type of optimization algorithm inspired from the principles and processes of the vertebrate immune system. In this paper, we propose a new approach to deal with the maximum satisfiability problem. It is a population based method where every individual represents a potential solution to the problem at hand. The features of the proposed method consist in applying different immune system principles like clonal selection and mutation to govern the dynamics of the population in a way to optimize a defined objective function. To foster the convergence to optimality, a local search has been embedded within the optimization process.

The remainder of the paper is organized as follows. In section 2, a formulation of the tackled problem is given. Section 3 presents some basic concepts of Artificial Immune Computing. In section 4, the proposed method is described. Experimental results are discussed in section 5. Finally, conclusions and future work are drawn.

2 Problem Formulation

Given a Boolean formula F expressed in CNF (Conjunctive Normal Form) and having n Boolean variables x_1, x_2, \dots, x_n , and m clauses. The k -SAT problem can be formulated as follows:

- An assignment to those variables is a vector $v = (v_1, v_2, \dots, v_n) \in \{0, 1\}^n$
- A clause C_i of length k is a disjunction of k literals, $C_i = (x_1 \text{ OR } x_2 \text{ OR } \dots \text{ OR } x_k)$
- Each literal is a variable or a negation of a variable
- Each variable can appear multiple times in the expression.

For some constant k , the k -SAT problem requests a variable assignment that makes a formula $F = C_1 \text{ AND } C_2 \text{ AND } \dots \text{ AND } C_m$ evaluate to true.

Max-SAT is the problem of finding the assignment that satisfies the highest possible number of clauses. Therefore, it is categorized as an optimization problem. The Max-SAT problem can be defined by specifying implicitly a pair (Ω, SC) , where Ω is the set of all potentials solution $(\{0, 1\}^n)$ and SC is a mapping $\Omega \rightarrow \mathbb{N}$, called score of the assignment, equal to the number of true clauses. Consequently, the problem consists of defining the best binary assignment that maximizes the number of true clauses in the Boolean formula. Clearly, there are 2^n potential satisfying assignments for this problem, and it has been proven that the k -SAT problem is NP-complete for any $k \geq 3$. There are other variances of the Max-SAT problem such as Weighted Max-SAT [9] and Partial Max-SAT [10].

In this paper we deal with the Max-3-SAT problem. It is a combinatorial optimization problem. Therefore, it is impossible to obtain exact solutions in polynomial time as the required computation grows exponentially with the size of the problem.

3 Artificial Immune Systems

A Natural Immune System (NIS) is a complex system that can be analyzed at different levels: molecules, cells and organs. Complex interactions between entities within each level enable the resulting immune system to protect the body from any harmful entity, or exogenous agent, called antigen. A specific kind of cells, known as B-cells, is responsible for the destruction of the antigen. The B-cell produces antibodies that binds with the antigens and marks them for destruction. The strength of the antibody/antigen binding is termed antigenic affinity [8].

One particular feature of the natural immune system is its ability to construct specific antibodies against new antigens. The clonal selection principle [8] handles this as follows: when a new antigen is detected, the available B-cells start producing antibodies. Those B-cells that best recognize the antigen proliferate by cloning. The clones then undergo hyper-mutation mechanisms to promote their genetic variation. B-cells with high affinity differentiate into plasma cells and memory cells, and B-cells with low affinity are either destroyed or mutated. Plasma cells produce a high number of antibodies against the invading antigen. Memory cells, on the other hand, are long living cells that confers the system a memory of the encountered antigen.

Artificial immune systems can be viewed as a composition of intelligent methodologies inspired from natural immune systems for solving real world problems [8]. Many models have been inspired from natural immune systems, such as negative selection and danger theory. Those models have been applied to a wide range of applications: multiple sequence alignment [11], network security, optimization and image alignment [12].

Artificial Clonal selection is an artificial immune system particularly adapted for optimization. Its basic algorithm is as follows [8]:

1. Generate a set of (**P**) candidate solutions, composed of the subset of memory cells (**Mc**) added to the remaining (**Pr**) population ($\mathbf{P} = \mathbf{Mc} + \mathbf{Pr}$);
2. Determine (Select) the n best individuals of the population (**Pn**) based on an affinity measure;
3. Reproduce (Clone) these best individuals of the population, giving rise to a temporary population of clones (**C**). The clone size is an increasing function of the affinity with the antigen;
4. Submit the population of clones to a hyper-mutation scheme, where the hyper-mutation is inversely proportional to the antigenic affinity of the antibody. A maturated antibody population is generated (**C***);
5. Re-select the improved individuals from **C*** to compose the memory set **Mc**. Some members of **P** can be replaced by other improved members of **C***;
6. Replace d antibodies by novel ones (diversity introduction). The lower affinity cells have higher probabilities of being replaced.

4 The Proposed Approach

To solve the Maximum Satisfiability problem, we propose a new algorithm, called ClonSAT, based on the clonal selection principles and enhanced by a local search procedure. ClonSAT starts from an initial random population of B-Cells; each B-cell

is an assignment and encodes a potential solution. Thus At each iteration, we begin by assessing the affinity of each B-cell. The affinity value is the number of satisfied clauses in the Boolean formula.

Next, only the N best B-cells are allowed to clone themselves. The better the B-cell, the more clones it will produce. The number of clones for each B-cell is calculated as follows:

$$nbClones_i = \frac{affinity_i}{\sum_j affinity_j} \times \beta \quad (1)$$

Where $affinity_i$ is the affinity of the i^{th} selected B-cell and β is a parameter that indicates the desired clone population's size.

Afterwards, the clone population starts a mutation process that transforms the clones to a degree inversely proportional to their affinity. The mutation process is as follows:

- a. For each clone $clone_i$; compute its normalized affinity:

$$affN_i = \frac{aff_i - \min Aff}{\max Aff - \min Aff} \quad (2)$$

$\min Aff$ and $\max Aff$ are the minimal and maximal affinities found in the $clone$ population.

- b. Calculate the number of mutations to apply to $clone_i$;

$$nbMutation_{s_i} = affN_i \times \min + (1 - affN_i) \times \max \quad (3)$$

The \min and \max parameters indicate how many mutations to apply to the worst and best individual respectively.

- c. For each mutation a random bit is flipped.

After the mutation step, we evaluate the affinity of the mature population and select the best B-cell as a candidate cell. If it is better than the memory cell, the candidate cell becomes the new memory cell. Finally, we replace all the B-cells of the current population with the best ones from both the current population and the mature population and we replace the worst elements of the current population with new randomly generated ones. The whole process is repeated until the stopping criterion is met.

In order to increase intensification capabilities of search, we apply a local search algorithm when the memory cell has not been improved for more than a given number of generations; for this purpose we propose the use of the well known WalkSat algorithm [13] as follows:

Start with the actual memory cell.

Repeat (predefined number of flips)

- Pick a random unsatisfied clause.
- Select and flip a variable from that clause:
 - i. With probability \mathbf{p} , pick a random variable
 - ii. With probability $\mathbf{1-p}$, pick greedily a variable that minimizes the number of unsatisfied clauses

5 Experimental Results

ClonSat has been implemented using Java language. To assess the experimental performance of ClonSat, we have performed several tests taken from the AIM benchmark instances. The AIM instances are all generated with a particular Random-3-SAT instance generator [14]. In all experiments, we have run the ClonSat program using the parameters' settings shown in Table 1.

Table 1. Parameters of ClonSAT

| Parameter name | Parameter value |
|---------------------------------------------|------------------|
| population size | 100 |
| mutation min | 1 / nb variables |
| mutation max | 0.01 |
| clone size (β) | 200 |
| replacement size | 10 |
| Total number of generations before stopping | 1000 |
| Nb generations before applying WalkSat | 100 |
| walkSat p probability | 0.2 |
| walkSat number of flips | 500 |

Furthermore, we have compared ClonSAT with the GSAT [6] and QSAT [15] algorithms. QSAT is a quantum evolutionary algorithm with a simple flipping procedure. Freidman tests were carried out to test the significance of the difference in the accuracy of each method in this experiment. The results, reported in table 2, are encouraging and prove the feasibility of using an artificial immune system to deal with

Table 2. Results

| Benchmark | Number of Variables (n) | Number of clauses (m) | QSAT | GSAT | ClonSAT alone | ClonSAT +WalkSat |
|--------------------|-------------------------|-----------------------|-----------|------------|---------------|------------------|
| Aim-50-1_6-no-1 | 50 | 80 | 79 | 79 | 79 | 79 |
| Aim-50-1_6-no-2 | 50 | 80 | 79 | 79 | 79 | 79 |
| Aim-50-1_6-no-3 | 50 | 80 | 78 | 79 | 79 | 79 |
| Aim-50-1_6-no-4 | 50 | 80 | 79 | 79 | 79 | 79 |
| Aim-200-2_0-no-1 | 200 | 400 | 396 | 399 | 399 | 399 |
| Aim-200-2_0-no-2 | 200 | 400 | 397 | 399 | 399 | 399 |
| Aim-200-2_0-no-3 | 200 | 400 | 397 | 399 | 399 | 399 |
| Aim-50-1_6-yes1-1 | 50 | 80 | 80 | 79 | 79 | 80 |
| Aim-50-1_6-yes1-2 | 50 | 80 | 80 | 79 | 79 | 79 |
| Aim-50-1_6-yes1-3 | 50 | 80 | 79 | 79 | 79 | 80 |
| Aim-100-2_0-no-1 | 100 | 200 | 198 | 199 | 199 | 199 |
| Aim-100-2_0-no-2 | 100 | 200 | 197 | 199 | 199 | 199 |
| Aim-100-3_4-yes1-3 | 100 | 340 | 335 | 335 | 336 | 340 |
| Aim-100-3_4-yes1-4 | 100 | 340 | 333 | 340 | 336 | 340 |

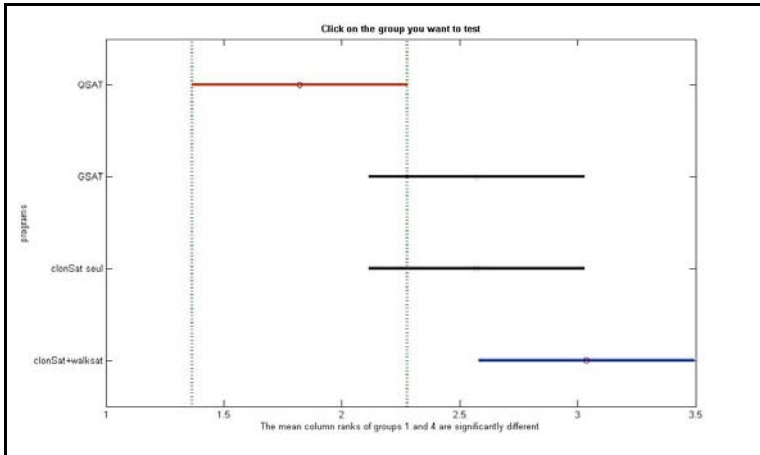


Fig. 1. Friedman test

the Max 3-Sat problem. In most cases, ClonSAT performs as well as GSAT and even better than QSAT, as it's shown by the Friedman test (figure 1). This is interesting because, apart from the affinity function, the algorithm that we used is a standard artificial immune system. The addition of walkSAT allows the program to perform even better with the most difficult cases.

6 Conclusion

In this paper we proposed a new approach, called ClonSAT, to solve the Max-3-SAT problem. ClonSAT is based on hybridizing an Artificial Immune System with a local search algorithm. Although more experiments and comparisons are required, the results so far are promising and demonstrate the feasibility of AISs to deal with the Max-SAT problem; in most cases, our program gives comparable or better solutions than GSAT and QSAT programs. In addition, the proposed framework provides an extensible platform for evaluating different variants of satisfiability problems. Our future work consists of investigating the effect of different local search methods on the performance of our approach.

References

1. Cook, S.A.: The Complexity of Theorem Proving Procedures. In: Proc. 3rd Ann. ACM Symp. On Theory of Computing, Association for Computing Machinery, pp. 151–158 (1971)
2. Marques-Silva, J.P., Sakallah, K.A.: GRASP: A Search Algorithm for Propositional Satisfiability. *IEEE Transactions on Computers* 48(5), 506–521 (1999)
3. Davis, M., Putnam, G., Loveland, D.: A machine program for theorem proving. *communication of the ACM*, 394–397 (1962)

4. Zhang, H., Shen, H.: Exact Algorithms for MAX-SAT. *Electronic Notes in Theoretical Computer Science* 86(1) (2003)
5. Marchiori, E., Rossi, C.: A Flipping Genetic Algorithm for Hard 3-SAT Problems. In: *Proc. of the Genetic and Evolutionary Computation Conference*, vol. 1, pp. 393–400 (1999)
6. Selman, B., Levesque, H., Mitchell, D.: A new method for solving hard satisfiability problems. In: *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI 1992)*, San Jose, CA, pp. 440–446 (1992)
7. Holger, H., Stützle, T.: Local search algorithms for SAT: An empirical evaluation. *Journal of Automated Reasoning* 24(4), 421–481 (2000)
8. De Castro, L.N.: The clonal selection algorithm with engineering applications. In: *Proc. GECCO, Workshop on Artificial Immune Systems*, pp. 36–37 (2000)
9. Borchers, B., Furman, J.: Two-Phase Exact Algorithm for MAX-SAT and Weighted MAX-SAT Problems. *Journal of Combinatorial Optimization* 2(4), 299–306 (1999)
10. Menai, M., Batouche, M.: A Backbone-Based Co-evolutionary Heuristic for Partial MAX-SAT. *Artificial Evolution*, 155–166 (2005)
11. Layeb, A., Deneche, A.: Multiple Sequence Alignment by Immune Artificial System. In: *IEEE/ACS International Conference on Computer Systems and Applications (AICCSA 2007)*, Jordan, pp. 336–342 (2007) ISBN: 1-4244-1031-2
12. Bendiab, E., Meshoul, S., Batouche, M.: An AIS for Multi-Modality Image Alignment. In: *Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS*, vol. 2787, pp. 13–21. Springer, Heidelberg (2003)
13. Selman, B., Kautz, H., Cohen, B.: Local Search Strategies for Satisfiability Testing. In: *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, October 11-13 (1993)
14. Asahiro, Y., Iwama, K., Miyano, E.: Random Generation of Test Instanzes with Controlled Attributes. In: *Johnson, D.S., Trick, M.A. (eds.) Cliques, Coloring, and Satisfiability: The Second DIMACS Implementation Challenge. DIMACS Series on Discr. Math. and Theor. Comp. Sci.*, vol. 26, pp. 377–394 (1996), <http://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/DIMACS/AIM/descr.html>
15. Layeb, A., Saidouni, D.: A New Quantum Evolutionary Local Search Algorithm for MAX 3-SAT Problem. In: *Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS (LNAI)*, vol. 5271, pp. 172–179. Springer, Heidelberg (2008)

Power-Aware Multi-objective Evolutionary Optimization for Application Mapping on NoC Platforms

Marcus Vinícius Carvalho da Silva¹, Nadia Nedjah¹, and Luiza de Macedo Mourelle²

¹ Department of Electronics Engineering and Telecommunications

² Department of System Engineering and Computation,
Engineering Faculty, State University of Rio de Janeiro, Brazil

Abstract. Network-on-chip (NoC) are considered the next generation of communication infrastructure, which will be omnipresent in different environments. In the platform-based design methodology, an application is implemented by a set of collaborating intellectual properties (IPs) blocks. The selection of the most suited set of IPs as well as their physical mapping onto the NoC to implement efficiently the application at hand are two hard combinatorial problems. In this paper, we propose an innovative power-aware multi-objective evolutionary algorithm to perform the assignment and mapping stages of a platform-based NoC design synthesis tool. Our algorithm can use one of the well-known multi-objective evolutionary algorithms NSGA-II and microGA as kernel. The optimization is driven by the required area and the imposed execution time considering that the decision maker's is the power consumption of the implementation.

1 Introduction

As the integration rate of semiconductors increases, more complex *system-on-chips* (SoCs) are launched. A simple SoC is formed by homogeneous or heterogeneous independent components while a complex SoC is formed by interconnected heterogeneous components. The interconnection and communication of these components by a communication architecture form a *network-on-chip* (NoC). A NoC is similar to a general network but with limited resources such as bandwidth, area and power. Each component of a NoC is designed as an *intellectual property* (IP) block.

Normally, a NoC is designed to run a specific application. This application, usually, consists of a limited number of tasks that are implemented by a set of IP blocks. An IP block can be assigned for more than a single task of the application or it can be dedicated to execute a single task. For instance, a processor IP block can execute different tasks as a general processor does but the NoC designer, due performance, can assign just one task for that specific processor. In the other hand, a multiplier IP block for floating point numbers can only multiply floating point numbers and the NoC designer can reuse that IP if the application has more than one task of floating point multiplication. The number of IP blocks designers, as well as the number of available IP blocks, is growing up fast.

A NoC consists of sets of *resources* and *switches*. Resources and switches are connected by *resource network interfaces*. Switches are connected by *communication channels*. The pair of switch/resource forms a *tile*. The simplest way to connect the available

resources and switches is arranging them as a mesh so these are able to communicate with each other by sending messages via an available communication path. A switch is able to buffer and route messages between resources. On a mesh-based NoC each switch is connected to up to four other neighboring switches through input and output channels. While a switch is sending data through a channel, it can buffer incoming data through another channel.

Usually, an application is described as a graph of tasks called *task graph* (TG), which is a high level description. The IP blocks features can be obtained from their manufacturer documentation. The IP assignment and IP mapping are key research problems for efficient NoC-based designs [7]. Electronic Design Automation (EDA) tools must deal with these two problems. On a low level description, every IP mapping must be synthesized, leading to a very slow but precise evaluation. On a high level description, evaluation is first driven by models of the NoC-based platform, leading to a fast evaluation and the precision depends of the modeling. Along the design process the description abstraction level is decreased until reach an RTL description.

IP assignment and IP mapping are combinatorial optimization problems classified as *NP-hard* problems [5]. We use multi-objective evolutionary algorithms (MOEAs) with specific operators and objective functions to yield an optimal IP assignment and IP mapping. As normally multi-objective problems present a set of solutions, we consider the preferences of a decision maker (DM) to find a single solution or at most a small subset of solutions. In this paper, we propose a power-aware multi-objective evolutionary decision support system to help NoC designers on a high level stage of a platform-based NoC design. For this purpose, we use two MOEAs: NSGA-II [2] and microGA [1]. Both of these algorithms were modified according to some prescribed NoC design constraints and to accept preferences defined by the DM.

The rest of the paper is organized as follows: In Section 2, we introduce the IP assignment and IP mapping problems in platform-based designs. Then, in Section 3, we describe a structured TG and IP repository model based on the E3S data. After that, in Section 4, we sketch the multi-objective evolutionary approach and present the objective functions. Later, in Section 5, we show some experimental result. Last but not least, in Section 6, we draw some conclusions and outline some future work.

2 IP Assignment and Mapping Problems

The platform-based design methodology for SoC encourages the reuse of components to reduce costs and to reduce the time-to-market of new designs. The designer of NoC-based systems faces two main problems: selecting the adequate set of IPs and finding the best physical mapping of these IPs into the NoC structure. On a platform-based design, the selection of IPs is called IP assignment stage and the physical mapping is called IP mapping stage.

The main objective of the IP assignment stage is to select, from a IP repository, a set of IPs that exploit re-usability and optimize the execution of a given application. At this stage, no information about physical location of IPs is available so optimization must be done based on application's description (as a TG) and IP features only. So, the result of this stage is the set of IPs that maximizes NoC performance due IPs features. The TG is

then annotated and an *application characterization graph* (ACG) is produced, wherein each node (task) has an IP assigned to it. The TG and ACG are defined in Section 3. The number of possible assignments is defined by $A = n_0 \times n_1 \times \dots \times n_{m-2} \times n_{m-1}$, wherein m represents the number of tasks in the application, t_0, t_1, \dots, t_{m-1} and n_i is the number of IPs that can be assigned to task t_i .

Given an application, described by its ACG, the problem that we are concerned with now is to determine how to topologically map the selected IPs onto the network platform, such that the objectives of interest are optimized. At this stage, a more accurate evaluation can be done taking into account the distance between resources and the number of switches and channels crossed by a data package along a path. The result of this process should be an optimal allocation of one of the prescribed IP assignments, to execute a desired application on a NoC platform.

The mapping stage uses the result obtained from the assignment, which consists of many non-dominated solutions. Let s be the number of distinct assignments evolved and p_i be the number of processors used in assignment i and n_i be the minimal number of resources in the NoC to be utilized in the implementation of the application with assignment solution i . In this case, the total number of possible mappings is defined as in (II).

$$M_s = \sum_{i=1}^s \frac{n_i!}{(n_i - p_i)!} \quad (1)$$

3 Task Graph and IP Repository Models

In order to formulate the IP mapping problem, it is necessary to introduce a formal definition of an application description first. An application can be described as a set of tasks that can be executed sequentially or in parallel. It can be represented by a directed acyclic graph of tasks, called *task graph*. A *Task Graph* (TG) $G = G(T, D)$ is a directed acyclic graph where each node represents a computational module in the application referred to as task $t_i \in T$. Each directed arc $d_{i,j} \in D$, between tasks t_i and t_j , characterizes either data or control dependencies.

Each task t_i is annotated with relevant information, such as a unique identifier and type of task in the network. Each $d_{i,j}$ is associated with a value $V(d_{i,j})$, which represents the volume of bits exchanged during the communication between tasks t_i and t_j . Once the IP assignment has been completed, each task is associated with an IP identifier. The result of the assignment is a graph of IPs representing the processor elements (PEs) responsible for executing the application. This graph is called *application characterization graph*. An ACG $G = G(C, A)$ is a directed graph, where each vertex $c_i \in C$ represents a IP assigned to one or more tasks, forming a core, and each directed arc $a_{i,j}$ characterizes the communication process from core c_i to core c_j . Each $a_{i,j}$ can be tagged with IP/application specific information, such as communication rate, communication bandwidth or volume of bits exchanged between cores c_i and c_j . A TG is based on application features only while an ACG is based on application and IP features, providing us with a much more realistic representation of an application in runtime on a NoC platform. The abstraction level decreases from the TG to the ACG representation.

4 Power-Aware Multi-objective Evolutionary Approach

The search space of the IP assignment and mapping problems, for a given application, may exceed millions or even billions of possible combinations. Among the huge number of possible solutions, it is possible to find many equally optimal solutions, called non-dominated solutions [1]. In huge non-continuous search space, deterministic approaches do not deal very well with MOPs. In order to deal with such a big search space in a reasonable time, a power-aware multi-objective evolutionary decision support system to aid platform-based NoC design is proposed.

Instead of obtain a single solution after IP assignment and IP mapping like in a typical platform-based NoC design, the proposed system exploits different solutions of assignment and mapping to equalizes the trade-off among the objectives of interest and introduces the DM's preferences to refine final result.

The kernel of the proposed aid system is driven by two well-known MOEAs: NSGA-II [2] and microGA [3]. Both adopt the domination concept with a ranking method of classification. In order to deal with the IP assignment and IP mapping problems both algorithms, i.e NSGA-II and microGA, were adapted to recognize two individuals representations: a assignment representation and a mapping representation. Originally, those algorithms do not consider the preferences of a DM and a major modifications were introduced to turn them into power-aware multi-objective genetic algorithms.

4.1 Representation

The chromosome is formed by a set of genes and each one represents a node *id* from the TG. Each gene *g* has a *IP id* field that corresponds to a IP from the repository capable to execute the associated task *type*. If a IP is dedicated to execute a single task of th TG, the *dedi* field value is 1, otherwise it is 0. Initially, a random IP *id* is assigned to each gene, with the constraint of the IP *type*. Tournament selection, one-point crossover and simple mutation were used. The crossover operator, without any constraint, can only produce feasible individuals because the order of genes is not changed. The mutation is controlled by IP *type* constraint to avoid selecting a random IP, from IP repository, of different *type*.

The mapping individual representation is inherited from the assignment individual representation. It is augmented with the *RES id* field, which indicates the resource on which a gene is mapped on the NoC platform, so representing a physical information. On a $N \times N$ regular mesh, assume that the tiles are numbered successively from top-left to bottom-right, row by row. The row of the i^{th} tile is given by $\lceil i/N \rceil$, and the corresponding column by $i \bmod N$. Note that the first resource *id* as the first row and the first column are numbered 0.

Three objectives of interest where identified for the platform-based NoC design optimization. In this paper the DM's preferences will constrain the power consumption of the NoC while the other two objectives, area occupied and time of execution, will be ranked as usually done.

4.2 Assignment Evaluation

The fitness of an assignment solution S is measured in terms of the silicon area that would be used to implement the NoC-based application using S , the approximate execution time of the so implemented application and the power consumption required when the application is executed. Note that in this case, only computation time and power due to computation are considered. Those introduced by the communication can only be considered when the actual location where the IPs are mapped within the NoC resource nodes are known, i.e. in the mapping stage. In the following, we explain in details how each of these characteristics is quantified.

Area. In order to compute the area required it is necessary to add up the area of each processor used in a given solution S . The identifier of each processor (*procID*) is retrieved visiting each gene of S . Grouping the nodes of same processor and identifying the nodes of dedicated processors, is a method to identify the processors of solution S . Equation 2 shows how to compute the area of solution S , wherein function $\mathcal{PE}(S)$ provides the set of non-dedicated processors used in S . The notation $S[t]_{ip}$ indicates the IP assigned to task t in S and $S[t]_{dedi}$ the value of field *dedi* for task t in S .

$$Area(S) = \sum_{t \in TG} area_{S[t]_{ip}} * S[t]_{dedi} + \sum_{p \in \mathcal{PE}(S)} area_p \quad (2)$$

Execution time. In order to compute the execution time required by a solution S , it is necessary to find the critical path of the ACG. The critical path can be found visiting all nodes of all paths and recording the execution time of the slowest path. When tasks that should be executed in parallel are allocated in the same processor, these tasks must be scheduled sequentially. If at least one of this tasks is from the critical path, the execution time will be increased. Assume that the scheduling order is dictated by the increasing order of the task identifier. In this context, consider the case where t_1, t_2, \dots, t_k , are k tasks that can be implemented in parallel, but are allocated to the same processor. The execution time associated with a path that goes through a task t_i is increased by the sum of execution times of all tasks that are scheduled before t_i . These tasks are those whose identifier is smaller than the identifier of the task t_i .

Equation 3 shows the details of this computation. In this context, the function $\mathcal{C}(g)$ returns all possible paths of the task graph g , function $\mathcal{P}(t)$ returns the set of all tasks in the ACG that may be executed in parallel with task t and are associated with the same processor in the solution S , function $\mathcal{D}(t)$ informs all the tasks that depends on the execution of t and that are also allocated to the same processor in S . Note that the attribute *level* of the nodes of a task graph can be used to determine the members of the set returned by function $\mathcal{P}(t)$.

$$Time(S) = \max_{M \in \mathcal{C}(GT)} \left(\sum_{t \in M} time_{S[t]_{ip}} + \mathcal{T}(S) \right)$$

$$\mathcal{T}(S) = \begin{cases} 0 & \text{if } S[t]_{dedi} = \mathbf{1or} \ \mathcal{P}(t) = \mathcal{D}(t) = \emptyset \\ \sum_{\substack{t' \in \mathcal{P}(t) \cup \mathcal{D}(t) \\ t' < t}} time_{S[t']_{ip}} & \text{otherwise} \end{cases} \quad (3)$$

Power consumption. To evaluate the power consumption of a application represented by a TG, the power consumption of each IP assigned must be added. In (4), $power_{S[t]_{ip}}$ represents the power consumption when a task t is executed by its assigned IP in a solution S , and ξ_a and ξ'_a are the power constraints imposed for the assignment.

$$\xi'_a \leq Power(S) = \sum_{t \in GT} power_{S[t]_{ip}} \leq \xi_a \quad (4)$$

4.3 Mapping Evaluation

The fitness of mapping solution S is measured in terms of the silicon area that would be used to implement the NoC-based application using S , the execution time of the so implemented application and the power consumption required when the application is executed. In the following, we explain in details how each of these characteristics is quantified.

Area. To compute the area required by a given mapping it is necessary to know the area needed for the selected processors and that occupied by the used channels and switches. As a processor can be responsible for more than one task, each ACG node must be visited in order to check the processor identifier for each node. It is necessary to identify those cases where a processor is dedicated for a task before grouping the nodes with same *procID* attribute. Nodes with same *procID* marked as non-dedicated are executed by the same processor. Nodes marked as dedicated are executed by a dedicated processor. The total number of channels and switches can be obtained through the consideration of all communication paths between exploited tiles. Note that a given IP mapping may not use all the available tiles, links and switches that are available in the NoC structure. Also, observe that a portion of a path may be re-used in several communication paths.

In this paper, we adopted XY deterministic route strategy [4]. The data emanating from tile i to j is sent first horizontally to the left or right side of the corresponding switch until it reaches the column of tile j , then, it is sent up or down, also depending on the position of tile j with respect to tile i until it reaches the row of tile j . The number of channels in the aforementioned route can be computed by the function $\mathcal{CH}(i, j)$ as described in (5). This also called the *Manhattan distance* between tiles i and j .

$$\mathcal{CH}(i, j) = |[i/N] - [j/N]| + |i \setminus N - j \setminus N| \quad (5)$$

The number of hops between tiles along a given path leads to the number of channels between those tiles, and incrementing that number by 1 yields the number of traversed switches, as shown in (6). The total area required is computed summing up the areas required by the implementation of all distinct processors, switches and channels. The area required by switches and channels depends of the NoC platform. A general decision support tool must allow the designer to configure these parameters for different platforms.

$$SW(i, j) = \mathcal{CH}(i, j) + 1 \quad (6)$$

Equation 7 describes the computation involved to obtain the total area of a given mapping solution S . For a given allocation, function $Area_A(\cdot)$ gives the area of the allocation exactly like (2) does. The allocation that originated mapping S is given by \mathcal{A}_S . Function $\mathcal{E}(g)$ returns all the edges of the task graph g , while attributes src and tgt returns the source and target tasks, respectively. Notation $S[t]_{res}$ indicates the resource's index where task t is mapped, regarding solution S . Constants $Area_c$ and $Area_s$ represents communication channel and switch areas, respectively.

$$Area_M(S) = Area_A(\mathcal{A}_S) + area_c \times \sum_{d \in \mathcal{E}(TG)} \mathcal{CH}(S[d_{src}]_{res}, S[d_{tgt}]_{res}) + area_s \times \sum_{d \in \mathcal{E}(TG)} \mathcal{SW}(S[d_{src}]_{res}, S[d_{tgt}]_{res}) \quad (7)$$

Execution time. To compute the execution time of a given mapping, we consider the execution time of each task of the critical path, their schedule and the additional time due to data transportation through channels and switches along the communication path. The execution time of each task is defined by the *taskTime* attribute in TG. Channels and switches can be counted using (5) and (6), respectively. Analyzing the assignment problem we identified a situation that increases the execution time of an application, which occurs when parallel tasks are allocated to the same processor. The mapping problem analysis revealed other two situations that can increase the execution time of the application: (i) Parallel tasks with common source sharing communication channels and (ii) Parallel tasks with common target sharing communication channels.

Equation 8 gives the execution time considering computation and communication for a given mapping solution S . Function $\mathcal{C}(g)$ returns all possible paths Q for a given task graph g , while $Time_p(Q)$ returns the time necessary to processes the tasks of a path Q and $Time_c(Q)$ the time spent due communication among tasks in path Q , assuming there is no contention. Considering contention, its necessary to add the delay concerning the two aforementioned situations. Delay caused by situation (i) is computed by function f_1 and delay caused by situation (ii) is computed by function f_2 in (8).

$$Time(S) = \max_{Q \in \mathcal{C}(GT)} (Time_p(Q) + Time_c(Q) + T'(Q)) \quad (8)$$

$$T'(Q) = t_L \times (f_1(Q) + f_2(Q))$$

The time spent due computation for a path Q of the task graph is computed as shown in (9). Function $\mathcal{P}(t)$ returns all the tasks at the same level of task t and associated with the same processor, for a given mapping solution S . Function $\mathcal{D}(t)$ returns all the task dependents of t and executed by the same processor, while $\mathcal{A}_S[t]_{ip}$ returns information about the IP assigned to a task t in S .

$$Time_p(Q) = \sum_{t \in Q} time_{\mathcal{A}_S[t]_{ip}} + T''(S)$$

$$T''(S) = \begin{cases} 0 & \text{if } \mathcal{A}_S[t]_{dedi} = 1 \text{ or } \mathcal{P}(t) = \mathcal{D}(t) = \emptyset \\ \sum_{\substack{t' \in \mathcal{P}(t) \cup \mathcal{D}(t) \\ t' < t}} time_{\mathcal{A}_S[t']_{ip}} & \text{otherwise} \end{cases} \quad (9)$$

The time spent due communication among tasks along a path Q is computed as shown in (10). Function $\mathcal{E}(TG)$ returns all the edges $d(src, tgt)$ of the task graph, $vol_{d(t, t')}$ is

the volume of bits transmitted from task t to task t' , t_R is the switch processing time and t_L is the channel transmission time.

$$Time_c(Q) = \sum_{\substack{d(t,t') \in \mathcal{E}(GT) \\ t \in Q, t' \in Q}} \left\lceil \frac{vol_{d(t,t')}}{phit} \right\rceil t_L SW(S[d_t]_{rec}, S[d_{t'}]_{rec})(t_R + t_L) \quad (10)$$

Function f_1 computes delays on path Q regarding situation (i). Algorithm 1 describes this function. For each parallel task that must be achieved through the same communication channel, the overall execution time is increased due to data pipelining. Function $Targets(t)$ returns all the tasks of the task graph that depends of task t , $i\mathcal{CH}(t, t')$ the initial communication channel index of task t to task t' and $penalty$ is the number of *flits* that would be transmitted when situation (i) occurs. A *flit* represents the flow unit, multiple of the *phit* that represents the physical unit given by the channel width. Function f_2 computes delays on path Q regarding to situation (ii). If a switch receives packages from two different channels at the same time and needs to route them through the same output channel, there will be a package pipelining. Algorithm 2 computes how many times this situation occurs. Function $\mathcal{CH}s(t, t')$ returns an ordered list of the required channels during communication of tasks t and t' and $penalty$ is the number of *flits* that would be transmitted when situation (ii) occurs.

Power consumption. To compute the power consumption of a mapping, it is necessary to consider the power consumed due processing and communication. The total power consumed is given as in (11), wherein $Power_p$ and $Power_c$ represent processing and communication consumption, respectively and ξ_m and ξ'_m the power constraints.

$$\xi'_m \leq Power(S) = Power_p(S) + Power_c(S) \leq \xi_m \quad (11)$$

Algorithm 1. $f_1(Q)$ – SameSrcDiffTgt

```

1:  $penalty := 0$ 
2: for all  $t \in Q$  do
3:   if  $Targets(t) > 1$  then
4:     Seja  $t_1 \in Targets(t) \mid t_1 \in Q$ 
5:     for all  $t_2 \in Targets(t) \setminus t_1$  do
6:       if  $i\mathcal{CH}(t, t_1) = i\mathcal{CH}(t, t_2)$  e  $t_1 > t_2$  then
7:          $penalty := penalty + \left\lceil \frac{vol(t, t_2)}{phit} \right\rceil$ 
8: return  $penalty$ 

```

Algorithm 2. $f_2(Q)$ – DiffSrcSameTgt

```

1:  $penalty := 0$ 
2: for all  $t \in Q$  do
3:   for all  $t_1 \in GT \mid t_1 \neq t$  e  $level(t) = level(t_1)$  do
4:     for all  $s \in Targets(t)$  e  $s_1 \in Targets(t_1) \mid s = s_1$  do
5:        $w = \mathcal{CH}s(t, s); w_1 = \mathcal{CH}s(t_1, s_1)$ 
6:       if exists  $i \in [0, \min(w.length, w_1.length)] \mid w(i) = w_1(i)$  e  $t > t_1$  then
7:          $penalty := penalty + \left\lceil \frac{vol(t_1, s)}{phit} \right\rceil$ 
8: return  $penalty$ 

```

The power consumption due processing is given summarizing the power consumption of each executed task of a mapping solution S . In (12), $Power_{(t,p)}$ represents the power consumed when a task t is executed by a processor p .

$$Power_p(S) = \sum_{t \in GT} power_{S[t]_{ip}} \quad (12)$$

The power consumed due communication is a important feature to be considered on a NoC power model in order to get a accurate evaluation. This feature depends on the application communication pattern and the NoC platform. The communication pattern is given by the assignment and mapping, while the NoC platform is defined by the network topology, switching strategy and routing algorithm. The power consumed by sending one bit from tile i to tile j is computed as shown in (13). Parameters $E_{S_{bit}}$ and $E_{C_{bit}}$ represents power consumed by switches and channels, respectively. These parameter are platform dependent and must be setted by the NoC designer.

$$E_{bit}^{i,j} = SW \times E_{S_{bit}} + CH \times E_{C_{bit}} \quad (13)$$

The task graph gives the volume of bits from task t to t' through oriented edge $d_{t,t'}$. Assuming that tasks t and t' are mapped on tiles i and j respectively, the amount of bits transmitted from tile i to j is denoted as $vol_{d(t,t')}$. Communication between tiles i and j can be established with a unique channel $C_{i,j}$ or with a sequence of $m > 1$ channels $[C_{i,x_0}, C_{x_0,x_1}, C_{x_1,x_2}, \dots, C_{x_{m-1},j}]$. For example, on a 3×3 mesh-based NoC with XY routing, a task mapped on tile 0 (top left) sends data to a task mapped on tile 8 (bottom right) through the following sequence of communication channels: $[C_{0,1}, C_{1,2}, C_{2,5}, C_{5,8}]$. The total power consumption due communication is given by (14), where $Targets(t)$ returns a set of tasks dependent of task t and $S[t]_{res}$ returns the tile index where task t is mapped, considering a mapping solution S .

$$Power_c(S) = \sum_{t \in TG, \forall t' \in Targets(t)} vol_{d(t,t')} \times E_{bit}^{S[t]_{res}, S[t']_{res}} \quad (14)$$

5 Results

The E3S (0.9) Benchmark Suite [3] was used to carry on the simulations. The suite contains the characteristics of 17 embedded processors. These processors are characterized by the measured execution times of 46 different types of tasks, power consumption derived from processor datasheets, die size required, price and clock frequency. In addition, E3S contains common applications executed by embedded systems in environments. We show the results obtained for 7 different applications, as described in the first 4 columns of Table 1, wherein N is the number of tasks and M is the number of data dependencies of the application. The minimal and average values of power consumption obtained were used in this paper to set the preferred minimal and maximal bounds, respectively, of power consumption for each application. However, any other value of power consumption can be set by the DM as suited.

Table 1. Applications and number of assignments and mappings for power-aware optimization

| ID | Application | N | M | Combinations | Assignment | | Mapping | |
|----|------------------------|-----|-----|--------------|------------|---------|---------|---------|
| | | | | | NSGA-II | microGA | NSGA-II | microGA |
| 1 | <i>auto-indust-tg0</i> | 6 | 4 | 1.183.744 | 2 | 4 | 2 | 7 |
| 2 | <i>auto-indust-tg2</i> | 9 | 9 | 606.076.928 | 17 | 23 | 11 | 47 |
| 3 | <i>consumer-tg0</i> | 7 | 8 | 2.247.264 | 9 | 6 | 3 | 10 |
| 4 | <i>consumer-tg1</i> | 7 | 5 | 176.868 | 3 | 9 | 7 | 18 |
| 5 | <i>networking-tg2</i> | 4 | 3 | 41.616 | 2 | 6 | 3 | 7 |
| 6 | <i>office-tg0</i> | 5 | 5 | 210.681 | 6 | 18 | 8 | 25 |
| 7 | <i>telecom-tg1</i> | 6 | 6 | 9.516.192 | 2 | 2 | 1 | 4 |

6 Conclusions

The problem of assigning and mapping IPs are NP-hard problems and key research problems in NoC design field. In this paper we propose a innovative power-aware multi-objective evolutionary decision support system to aid NoC designers assigning and mapping a prescribed set of IPs into a NoC physical structure. A power-aware optimization was performed and the performance of the NSGA-II and microGA compared. The latter performed better for all applications.

References

1. Coello, C.A.C., et al.: Evolutionary Algorithms for Solving Multi-Objective Problems. In: Genetic and Evolutionary Computation. Springer, Heidelberg (2006)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 182–197 (2002)
3. Dick, R.P.: Embedded System Synthesis Benchmarks Suite, E3S (2008)
4. Duato, J., Yalamanchili, S., Ni, L.: Interconnection Networks: An Engineering Approach. Morgan Kaufmann, San Francisco (2003)
5. Garey, M.R., Johnson, D.S.: Computers and intractability; a guide to the theory of NP-completeness. W. H. Freeman, USA (1979)
6. Murali, S., De Micheli, G.: SUNMAP: a tool for automatic topology selection and generation for nocs. In: Proc. of DAC 2004, pp. 914–919. ACM Press, New York (2004)
7. Ogras, Ü.Y., et al.: Key research problems in NoC design: a holistic perspective. In: Proc. Conf. on HW/SW Codesign and System Synthesis, pp. 69–74. ACM Press, New York (2005)

A Discrete Differential Evolution Algorithm for Solving the Weighted Ring Arc Loading Problem

Anabela Moreira Bernardino¹, Eugénia Moreira Bernardino¹,
Juan Manuel Sánchez-Pérez², Juan Antonio Gómez-Pulido²,
and Miguel Angel Vega-Rodríguez²

¹ Research Center for Informatics and Communications, Department of Computer Science, School of Technology and Management, Polytechnic Institute of Leiria, 2411 Leiria, Portugal
{anabela.bernardino, eugenia.bernardino}@ipleiria.pt

² Department of Technologies of Computers and Communications, Polytechnic School, University of Extremadura, 10071 Cáceres, Spain
{sanperez, jangomez, mavega}@unex.es

Abstract. Resilient Packet Ring is a recent telecommunication transport technology that combines the appealing functionalities from Synchronous Optical Network/ Synchronous Digital Hierarchy networks with the advantages of Ethernet networks. To effectively use the RPR's potential, namely the spatial reuse, statistical multiplexing and bi-directionality, it is necessary to route the demands efficiently. Given a set of point-to-point unidirectional traffic demands of a specified bandwidth, the demands should be assigned to the clockwise or to the counter-clockwise ring in order to yield the best performance. This paper suggests an efficient load balancing algorithm - Discrete Differential Evolution. We compare our results with the ones obtained by the Genetic Algorithm, the Differential Evolution, the Tabu Search and the Particle Swarm Optimisation, used in literature. The simulation results verify the effectiveness of the DDE.

Keywords: Ring Networks, Optimisation Algorithms, Differential Evolution Algorithm, Weighted Ring Arc Loading Problem.

1 Introduction

Resilient Packet Ring (RPR), also known as IEEE 802.17, is a standard designed for optimising the transport of data traffic over optical fibre ring networks [1-3]. The load balancing model for RPR differs from the Synchronous Optical Network/ Synchronous Digital Hierarchy (SONET/SDH) ring loading. Namely, in SONET/SDH rings the demands assigned to go clockwise compete for common span capacity with the demands assigned to go counter-clockwise. In RPR two distinct rings occur and the demands do not compete for the common capacity. The Weighted Ring Arc Loading Problem (WRALP) arises in the engineering and planning of the RPR systems, while the Weighted Ring Edge Loading Problem (WRELP) arises in the SONET/SDH rings. The load of an arc is defined to be the total weight of those requests that are routed through the Arc in its direction (WRALP) and the load of an edge is the number of routes traversing the Edge in either direction (WRELP). WRALP/WRELP ask for such a routing scheme that the maximum load on arcs/edges will be minimum.

The load balancing problems can be classified into two formulations: with demand splitting (split) or without demand splitting (non-split). The split loading allows the splitting of a demand into two portions to be carried out in both directions, while in a non-split loading each demand must be entirely carried out in either clockwise or counter-clockwise direction. In this paper we study the non-split WRALP.

Researchs on the no-split WRELP performed by Cosares and Saniee [4] and Dell'Amico et al. [5] studied the problem on SONET rings. Cosares and Saniee [4] proved that the formulation without demand splitting is a NP-complete problem. Recent studies on the non-split WRELP use Evolutionary Algorithms (EAs) [6][7]. For the split WRELP, Schrijver et al. [8] summarise various approaches and their algorithms are compared in Myung and Kim [9] and Wang [10].

The WRALP considered in the present paper is identical to the one described by Kubat and Smith [11] (non-split), Cho et al. [12] (split and non-split) and Yuan and Zhou [13] (split). They try to find approximate solutions in a reduced amount of time. Our purpose is different, we want to compare the performance of our algorithm with others in the achievement of the best-known solution. Using the same principle Bernardino et al. [14] proposed four hybrid Particle Swarm Optimisation (PSO) algorithms to solve the non-split WRALP.

In this paper we propose a Discrete Differential Evolution (DDE) algorithm to solve the non-split WRALP. Our algorithm is based on the DDE algorithm proposed by Pan et al. [15] for solving the permutation flowshop scheduling problem. The DDE algorithm first mutates a target population to produce the mutant population. Then the target population is recombined with the mutant population in order to generate a trial population. Finally, a selection operator is applied to both target and trial populations to determine who will survive for the next generation [15]. We use a Local Search (LS) embedded in the DDE algorithm to improve the solution quality.

We compare the performance of DDE with four algorithms: Genetic Algorithm (GA), Differential Evolution (DE), Tabu Search (TS) and LS-Probability Binary PSO (LS-PBPSO), used in literature.

The paper is structured as follows. In Section 2 we describe the WRALP; in Section 3 we present the DDE algorithm; in Section 4 we discuss the computational results obtained and, finally, in Section 5 we report about the conclusions.

2 WRALP Definition

Let R_n be a n -node bidirectional ring with nodes $\{n_1, n_2, \dots, n_n\}$ labelled clockwise. Each edge $\{e_k, e_{k+1}\}$ of R_n , $1 \leq k \leq n$ is taken as two arcs with opposite directions, in which the data streams can be transmitted in either direction: $a_k^+ = (e_k, e_{k+1})$, $a_k^- = (e_{k+1}, e_k)$. A communication request on R_n is an ordered pair (s, t) of distinct nodes, where s is the source and t is the destination. We assume that data can be transmitted clockwise or counter-clockwise on the ring, without splitting. We use $P^+(s, t)$ to indicate the directed (s, t) path clockwise around R_n , and $P^-(s, t)$ the directed (s, t) path counter-clockwise around R_n .

A request (s, t) is often associated with an integer weight $w \geq 0$; we denote this weighted request by $(s, t; w)$. Let $D = \{(s_1, t_1; w_1), (s_2, t_2; w_2), \dots, (s_m, t_m; w_m)\}$ be a set of integrally weighted requests on R_n . For each request/pair

(s_i, t_i) we need to design a directed path P_i of R_n from s_i to t_i . A set $P = \{P_i : i=1, 2, \dots, m\}$ of such directed paths is called a routing for D .

In this work, the solutions are represented using binary vectors (Table 1). For some integer $V_i = 1, 1 \leq i \leq m$, the total amount of data is transmitted along $P^+(s_i, t_i)$; $V_i=0$, the total amount of data is transmitted along $P^-(s_i, t_i)$. The vector $V = (V_1, V_2, \dots, V_m)$ determines a routing scheme for D .

Table 1. Solution representation

| | | | | | | |
|--------------------|-------------------|-------------------|---------------------|-------------------|-------------------|-------------------|
| Pair(s,t) Demand | C-clockwise | | CC-counterclockwise | | | |
| 1: (1, 2) → 15 | 15 | C | | | | |
| 2: (1, 3) → 3 | 3 | CC | | | | |
| 3: (1, 4) → 6 | 6 | CC | | | | |
| 4: (2, 3) → 15 | 15 | C | | | | |
| 5: (2, 4) → 6 | 6 | CC | | | | |
| 6: (3, 4) → 14 | 14 | C | | | | |
| Representation (V) | Pair ₁ | Pair ₂ | Pair ₃ | Pair ₄ | Pair ₅ | Pair ₆ |
| | 1 | 0 | 0 | 1 | 0 | 1 |

3 The Proposed Discrete Differential Evolution Algorithm

The DE was introduced by Storn and Price in 1995 [16]. The DE algorithm is a population-based algorithm using crossover, mutation and selection operators [17]. It resembles the structure of an EA, but differs in the generation of new candidate solutions and by using a ‘greedy’ selection scheme. The DE algorithm uses the mutation operation as a search mechanism and the selection operation to direct the search toward the prospective regions in the search space. The algorithm also uses a non-uniform crossover. By using the components of the existing population members to build trial vectors, the crossover operator efficiently shuffles information about successful combinations, enabling the search for a better solution space [17].

The application of the DE on combinatorial optimisation problems with binary decision variables is not usual. One of the possible reasons is the encoding scheme. Most of the discrete problems have solutions obtainable through permutation vectors, while the DE maintains and evolves floating-point vectors. In order to apply the DE to solve binary problems, the most important is to find a suitable encoding scheme, which can perform a transformation between floating-point and permutation vectors.

Bernardino et al. [7] proposed a DE (HDE) algorithm to solve the WREL P. The HDE algorithm applies a separate LS process to improve individuals. It combines global and local search by using an EA to perform exploration, while the LS method performs exploitation. The HDE algorithm uses the binary representation. After applying the standard equations, the algorithm verifies if the trial solutions contain values outside the allowed range. If a gene (pair) is outside of the allowed range, it is necessary to apply the following transformation:

| | | | |
|--------------------|----------|----------------------|----------|
| IF pair \geq 0.5 | pair = 1 | ELSE IF pair $<$ 0.5 | pair = 0 |
|--------------------|----------|----------------------|----------|

This transformation significantly increases the algorithm execution time. To solve this problem we use a DDE algorithm [15]. The solutions are based on discrete values and can be applied to all types of combinatorial optimisation problems [18].

The main steps of the DDE algorithm are given below:

```

Initialise Parameters
Create initial Population, P0
Evaluate Population P0
Find Best Solution in P0, Pg
WHILE stop criterion is not reached
    Create Mutant Population, Pmt
    Create Trial Population, Ptt
    Evaluate Trial Population Ptt
    Make Selection and update target population, Pt
    Find Best solution in Pt, Pgt
    Apply Local Search to Pgt
    
```

Initialization of Parameters

The following parameters must be defined by the user: (1) ni - number of individuals; (2) mi - maximum number of iterations; (3) pp - perturbation probability; (4) np - number of perturbations and (5) pc - crossover probability.

Target Population

The initial solutions can be created randomly or in a deterministic form. The deterministic form is based in a Shortest-Path Algorithm (SPA). The SPA is a simple traffic demand assignment rule in which the demand will traverse the smallest number of segments.

Evaluation of Solutions

To evaluate how good a potential solution is in relation to other potential solutions, we use the following fitness function:

$$w_i, \dots, w_m \rightarrow \text{demands of the pairs } (s_i, t_i), \dots, (s_m, t_m) \tag{1a}$$

$$V_i, \dots, V_m = 0 \rightarrow P^-(s_i, t_i); 1 \rightarrow P^+(s_i, t_i) \tag{1b}$$

$$\text{Load on arcs: } L(V, a_k^+) = \sum_{i: a_k^+ \in P^+(s_i, t_i)} w_i \quad L(V, a_k^-) = \sum_{i: a_k^- \in P^-(s_i, t_i)} w_i \tag{2a}$$

$$\forall k=1, \dots, n; \quad \forall i=1, \dots, m \tag{2b}$$

$$\text{Fitness function: } \max\{\max L(V, a_k^+), \max L(V, a_k^-)\} \tag{3}$$

The fitness function is responsible for performing the evaluation and returning a positive number (fitness value) that reflects how optimal the solution is. It is based on the following constraints: (1) between each node pair (s_i, t_i) there is a demand value >=0. Each positive demand value is routed in either clockwise (C) or counter-clockwise (CC) direction; (2) for an arc the load is the sum of w_k for clockwise or counter-clockwise direction between nodes e_k and e_{k+1}. The objective is to minimise the maximum load on the arcs of a ring (3).

Mutant Population

A mutant individual is obtained by perturbing the best solution of previous generation in the target population. To obtain the mutant individual, the following equation is

used: $Pm_i^t = \begin{cases} DC_{np}(P_g^{t-1}) & \text{if } (r < pp) \\ \text{MutationCbsestConcentrator}(P_g^{t-1}) & \end{cases}$

P_g^{t-1} is the best solution from the previous generation in the target population and DC_{np} is the destruction and construction procedure with the destruction size of np as a perturbation operator; and *MutationDirection* is a simple mutation operator. With this operator, one gene (pair) is randomly selected and its direction is exchanged. A uniform random number r is generated between 0 and 1. If r is less than p_p then the perturbation operator (DC) is applied to generate the mutant individual: $Pm_i^t = DC_{np}(P_g^{t-1})$. Otherwise, the best solution from the previous population is perturbed using a simple mutation operator: $Pm_i^t = MutationDirection(P_g^{t-1})$.

The general mechanism of the DC procedure is represented below:

```

FOR i=1 TO np DO
  i1= random (m/2)           i2= random (m-m/2)
  for c=i1 to c=i2+i1
    if random(2)=0
      if (sc- tc) = n/2
        if random(2)=1 testSolution[c] = CC
        else testSolution[c] = C
      else if (sc- tc) > n/2 testSolution[c] = C
      else testSolution[c] = CC
    else testSolution[c]= bestSolution[c];
    if fitnessTest < fitnessOld break;
  if fitnessTest < fitnessNew newSolution=testSolution

```

A new solution is obtained by performing multiple perturbations (np). Some of the pairs of the solution are selected and in 50% of the cases their direction is changed to the shortest path; otherwise it changes its direction to the direction of the best solution of the population. The algorithm repeats this process until the np is reached.

Trial Population

Following the perturbation phase, a trial individual is obtained using the following expression: $Pt_i^t = \begin{cases} CR(Pm_i^t, P_i^{t-1}) & \text{if } (r < p_c) \\ Pm_i^t & \end{cases}$. A uniform random number r is generated between 0 and 1. If r is lower than p_c then the crossover operator is applied to generate the trial individual: $Pt_i^t = CR(P_m^t, P_i^{t-1})$. The crossover operator CR adopted was ‘‘Uniform’’[6]. The crossover operator produces two children. In this study, we selected one of the children randomly. If r is higher or equal to p_c then the trial individual is chosen as: $Pt_i^t = P_i^{t-1}$. The trial individual is made up either from the perturbation operator or from the crossover operator.

Selection

The selection is based on the survival of the fitness among the trial and target individuals using the following expression: $P_i^t = \begin{cases} Pt_i^t & \text{if } (fitness(Pt_i^t) < fitness(P_i^{t-1})) \\ P_i^{t-1} & \end{cases}$.

Local Search

The LS algorithm applies a partial neighbourhood examination. We create two different LS methods that can be chosen by the user. In the LS “Exchange Direction” (LS-ED) some pairs of the solution are selected and their directions are exchanged (partial search). This method can be summarised in the following pseudo-code steps:

```

For t=0 to numberNodesRing/4
  P1 = random (number of pairs)           P2 = random (number of pairs)
  N = neighborhoods of ACTUAL-SOLUTION (one neighborhood results of
interchange the direction of P1 and/or P2)
SOLUTION = FindBest (N)
If ACTUAL-SOLUTION is worst than SOLUTION
  ACTUAL-SOLUTION = SOLUTION

```

In the LS “Exchange Max Arc” (LS-EMA), first it is necessary to establish the arc with the highest fitness. A set of neighbours is obtained by interchanging the direction of some pairs that flow by the arc with the highest fitness (partial search). This method can be summarised in the following pseudo-code steps:

```

Define MaxArc                               y1= random(m/2)           y2= random(m/2)
FOR p=y1 TO y1+y2
  IF p flows by MaxArc
    N = neighborhoods of ACTUAL-SOLUTION (one neighborhood results of
Interchange the direction of p)
SOLUTION = FindBest (N)
If ACTUAL-SOLUTION is worst than SOLUTION
  ACTUAL-SOLUTION = SOLUTION

```

Termination Criterion

The algorithm stops when a maximum number of iterations (mi) is reached.

Further information on DDE can be found in [19].

4 Results

We evaluate the utility of the algorithms using the same examples produced by Bernardino et al. [14]. They consider six different ring sizes: 5, 10, 15, 20, 25 and 30 and four demand cases: (1) complete set of demands between 5 and 100 with uniform distribution; (2) half of the demands in (1) set to zero; (3) 75% of the demands in (1) set to zero and (4) complete set of demand between 1 and 500 with uniform distribution. The last case was only used for the 30 nodes ring. For convenience, the instances used are labeled C_{ij} , where $1 < i < 6$ represents the ring size and $1 < j < 4$ represents the demand case.

We perform comparisons between all parameters of the DDE using the instance C41 with 50 iterations and creating the initial solutions randomly, to obtain the best combination of parameters.

The best results obtained with DDE use n_p between 5 and 10, $p_p > 0.6$ and $p_c < 0.3$ (Fig. 1) and $n_i = [50, 100]$. These parameters were experimentally found to be good and robust for the problems tested.

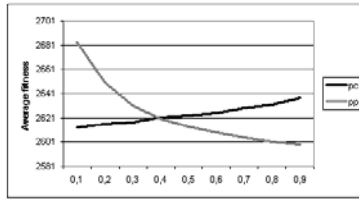


Fig. 1. Influence of parameters

We studied the influence of the two different LS methods developed on the execution time, the average fitness and the number of best solutions found. The LS-ED obtains a better average fitness, however the LS-EMA is less time consuming. We verify that at the same time, no matter the number of iterations, the two LS methods produce an identical number of best solutions (Fig. 2 and Fig. 3).

In our experiments we use different population sizes. The number of individuals was set to $\{10, 20, 30, \dots, 160\}$. We studied the impact on the execution time, the average fitness and the number of best solutions found (Fig. 2). The best values are between 50 and 100. With these values the algorithm can reach a reasonable number of good solutions in a reasonable amount of time (Fig. 2c). With a higher number of initial solutions, the algorithm can reach a better average fitness, but it is more time consuming (Fig. 2b).

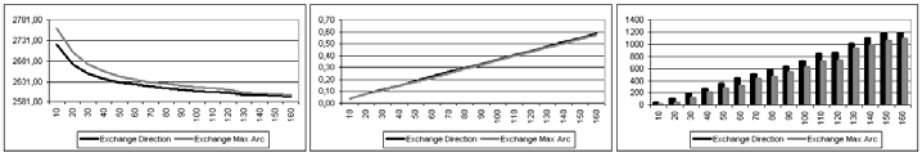


Fig. 2. Number of Individuals – Average Fitness (a) / Execution Time (b) / Best Solutions (c)

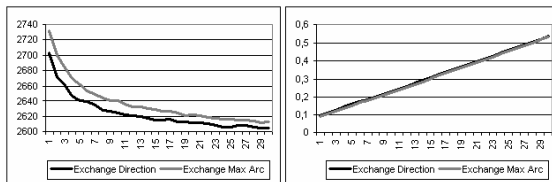


Fig. 3. Number of Perturbations – Average Fitness (a) / Average Time (b)

For parameter np , the number of perturbations, np between 5 and 10 has been shown experimentally to be more efficient (Fig. 3). In our experiments np was set to $\{1, 2, 3, \dots, 30\}$. A small np did not allow the system to escape from local minima, because the resulting solution was in most cases the same as the starting permutation. A high np has a significant impact on the execution time (Fig. 3b).

In general, experiments have shown that the proposed parameter setting for the DDE is very robust to small modifications.

To compare our results we consider the results produced with the GA proposed by Bernardino et al. [6], the DE and TS algorithms proposed by Bernardino et al. [7], and the LS-PBPSO proposed by Bernardino et al. [14]. The suggestions from literature helped to guide our choice of parameter values for the GA [6], DE and TS algorithms [7] and the LS-PBPSO algorithm [14]. The GA was applied to populations of 200 individuals, it uses the “Uniform” method for recombination, the “Change Direction” method for mutation and the “Tournament with Elitism” for selection. For the GA, we consider crossover probability in the range [0.6, 0.9] and a mutation probability in the range [0.5, 0.7]. The DE was applied to populations of 50 individuals, it uses the strategy “Best1Bin”, CR in the range [0.3, 0.5] and factor F in the range [0.5, 0.7]. The TS considers a number of elements in the tabu list between 4 and 10. The LS-PBPSO was applied to populations of 40 particles and we consider the value 1.49 for the parameters C1 and C2, and for the inertia velocity (w) values in the range [0.6, 0.8]. For the DDE we consider populations of 50 individuals, 5 perturbations, p_c in the range [0.1, 0.2], p_p in the range [0.6, 0.8] and the LS method “Exchange Direction”.

The algorithms have been executed using a processor Intel Quad Core Q9450 and the initial solutions of all algorithms were created using random solutions. For the problem C64 we used the SPA for creating the initial populations.

Table 2 presents the best obtained results by Bernardino et al. [14]. The first column represents the number of instance (Instance), the second and the third columns show the number of nodes (Nodes) and the number of pairs (Pairs) and finally the fourth column shows the minimum fitness values obtained.

Table 2. Best obtained results

| Instance | Nodes | Pairs | Best Fitness | Instance | Nodes | Pairs | Best Fitness |
|----------|-------|-------|--------------|----------|-------|-------|--------------|
| C11 | 5 | 10 | 161 | C41 | 20 | 190 | 2581 |
| C12 | 5 | 8 | 116 | C42 | 20 | 93 | 1482 |
| C13 | 5 | 6 | 116 | C43 | 20 | 40 | 612 |
| C21 | 10 | 45 | 525 | C51 | 25 | 300 | 4265 |
| C22 | 10 | 23 | 243 | C52 | 25 | 150 | 2323 |
| C23 | 10 | 12 | 141 | C53 | 25 | 61 | 912 |
| C31 | 15 | 105 | 1574 | C61 | 30 | 435 | 5762 |
| C32 | 15 | 50 | 941 | C62 | 30 | 201 | 2696 |
| C33 | 15 | 25 | 563 | C63 | 30 | 92 | 1453 |
| | | | | C64 | 30 | 435 | 27779 |

Table 3 presents the best results obtained with GA, HDE, TS, LS-PBPSO and DDE. The first column represents the number of the problem (Prob), the second column demonstrates the number of iterations used to test each instance and the remaining columns show the results obtained (Time – Run Times, IT - Iterations) by the five algorithms. The results have been computed based on 100 different executions for each test instance, using the best combination of parameters found and different seeds. Table 3 considers only the 30 best executions. All the algorithms reach the best solutions before the run times and number of iterations presented.

Table 3. Results – run times and number of iterations

| Prob | Number Iterations | GA | | HDE | | Tabu Search | | LS-PBPSO | | DDE | |
|------|-------------------|--------|-----|--------|----|-------------|-----|----------|-----|--------|----|
| | | Time | IT | Time | IT | Time | IT | Time | IT | Time | IT |
| C11 | 25 | <0.001 | 2 | <0.001 | 2 | <0.001 | 5 | <0.001 | 2 | <0.001 | 2 |
| C12 | 10 | <0.001 | 2 | <0.001 | 2 | <0.001 | 5 | <0.001 | 2 | <0.001 | 2 |
| C13 | 10 | <0.001 | 1 | <0.001 | 1 | <0.001 | 1 | <0.001 | 1 | <0.001 | 1 |
| C21 | 50 | <0.001 | 15 | <0.001 | 10 | <0.001 | 25 | <0.001 | 15 | <0.001 | 10 |
| C22 | 25 | <0.001 | 5 | <0.001 | 3 | <0.001 | 5 | <0.001 | 3 | <0.001 | 3 |
| C23 | 10 | <0.001 | 3 | <0.001 | 3 | <0.001 | 5 | <0.001 | 3 | <0.001 | 3 |
| C31 | 100 | 0.1 | 30 | 0.1 | 15 | 0.1 | 90 | 0.1 | 20 | 0.1 | 10 |
| C32 | 50 | <0.001 | 15 | <0.001 | 5 | <0.001 | 30 | <0.001 | 8 | <0.001 | 5 |
| C33 | 25 | <0.001 | 5 | <0.001 | 3 | <0.001 | 20 | <0.001 | 5 | <0.001 | 3 |
| C41 | 300 | 0.1 | 60 | 0.1 | 30 | 0.3 | 220 | 0.1 | 40 | 0.08 | 25 |
| C42 | 100 | 0.075 | 40 | 0.05 | 10 | 0.1 | 85 | 0.05 | 20 | 0.03 | 8 |
| C43 | 50 | <0.001 | 10 | <0.001 | 5 | <0.001 | 25 | <0.001 | 5 | <0.001 | 3 |
| C51 | 500 | 0.75 | 80 | 0.75 | 30 | 1 | 260 | 0.75 | 80 | 0.5 | 30 |
| C52 | 400 | 0.1 | 40 | 0.1 | 15 | 0.2 | 110 | 0.1 | 25 | 0.08 | 15 |
| C53 | 250 | 0.01 | 25 | 0.01 | 10 | 0.03 | 200 | 0.01 | 15 | 0.005 | 8 |
| C61 | 1500 | 1,75 | 130 | 1.75 | 40 | 3.5 | 400 | 2 | 130 | 1.5 | 50 |
| C62 | 1000 | 0.2 | 60 | 0.25 | 20 | 0.5 | 230 | 0.4 | 50 | 0.15 | 25 |
| C63 | 500 | 0.075 | 30 | 0.075 | 10 | 0.1 | 100 | 0.075 | 15 | 0.05 | 10 |
| C64 | 500 | 0.3 | 30 | 0.25 | 3 | 1.5 | 250 | 0.75 | 40 | 0.1 | 3 |

The DDE, HDE and GA obtain better solutions for larger instances. The TS is the slowest algorithm and it obtains a higher average fitness (Table 4). DDE is the faster algorithm for larger instances.

When using the SPA for creating the initial solutions, the times and number of iterations decreases – problem C64. This problem is computationally harder than the C61, however the best solution is obtained faster. To improve the solutions we consider more efficient to apply initially a SPA and then the meta-heuristic to improve the solutions.

Table 4 presents the average fitness and the average time obtained with GA, HDE, TS, LS-PBPSO and DDE using a limited number of iterations for the problems C41, C51 and C61 (harder problems). The first column represents the number of the problem (Prob), the second column demonstrates the number of iterations used to test each instance and the remaining columns show the results obtained (AvgF – Average Fitness, AvgT – Average Time) by the five algorithms. The results have been computed based on 100 different executions for each test instance using the best combination of parameters found and different seeds.

Table 4. Results – Average Time / Average Fitness

| Problem | Number of iterations | GA | | HDE | | LS-PBPSO | | Tabu | | DDE | |
|---------|----------------------|---------|------|---------|------|----------|------|---------|------|---------|------|
| | | AvgF | AvgT | AvgF | AvgT | AvgF | AvgT | AvgF | AvgT | AvgF | AvgT |
| C41 | 50 | 2587,62 | 0,17 | 2584,31 | 0,27 | 2594,36 | 0,26 | 2635,28 | 0,16 | 2582,06 | 0,12 |
| C51 | 75 | 4273,18 | 0,43 | 4271,27 | 0,71 | 4291,52 | 0,86 | 4392,70 | 0,86 | 4268,96 | 0,47 |
| C61 | 100 | 5785,62 | 1,34 | 5783,18 | 1,87 | 5837,58 | 3,10 | 5963,14 | 3,71 | 5781,52 | 1,27 |

The DDE is the algorithm that presents the best average fitness in the best execution time.

5 Conclusions

In this paper we present a DDE algorithm to solve the WRALP. The performance of our algorithm is compared with the algorithms: GA, HDE, TS and LS-PBPSO.

Relatively to the problem studied, the DDE algorithm presents better results. The DDE provides good solutions in a smaller execution time.

Experimental results demonstrate that the proposed DDE algorithm is an effective and competitive approach in composing fairly satisfactory results regarding the solution quality and execution time for the WRALP. When using the SPA for creating the initial solutions, the best solution is obtained faster.

In literature the application of the DDE for this problem is nonexistent, for that reason this article shows its enforceability in the resolution of this problem.

The continuation of this work will be the search and implementation of new methods for speeding up the optimisation process.

References

1. RPR Alliance: A Summary and Overview of the IEEE 802.17 Resilient Packet Ring Standard (2004)
2. Davik, F., Yilmaz, M., Gjessing, S., Uzun, N.: IEEE 802.17 Resilient Packet Ring Tutorial. *IEEE Communications Magazine* 42(3), 112–118 (2004)
3. Yuan, P., Gambiroza, V., Knightly, E.: The IEEE 802.17 Media Access Protocol for High-Speed Metropolitan-Area Resilient Packet Rings. *IEEE Network* 18(3), 8–15 (2004)
4. Cosares, S., Saniee, I.: An optimization problem related to balancing loads on SONET rings. *Telecommunication Systems* 3(2), 165–181 (1994)
5. Dell’Amico, M., Labbé, M., Maffioli, F.: Exact solution of the SONET Ring Loading Problem. *Oper. Res. Lett.* 25(3), 119–129 (1999)
6. Bernardino, A.M., Bernardino, E.M., Sánchez-Pérez, J.M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A.: Solving the Ring Loading Problem using Genetic Algorithms with intelligent multiple operators. In: *International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008)*, pp. 235–244. Springer, Heidelberg (2008)
7. Bernardino, A.M., Bernardino, E.M., Sánchez-Pérez, J.M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A.: Solving the weighted ring edge-loading problem without demand splitting using a Hybrid Differential Evolution Algorithm. In: *The 34th IEEE Conference on Local Computer Networks*. IEEE Press, Los Alamitos (2009)
8. Schrijver, A., Seymour, P., Winkler, P.: The ring loading problem. *SIAM Journal of Discrete Mathematics* 11, 1–14 (1998)
9. Myung, Y.S., Kim, H.G.: On the ring loading problem with demand splitting. *Operations Research Letters* 32(2), 167–173 (2004)
10. Wang, B.F.: Linear time algorithms for the ring loading problem with demand splitting. *Journal of Algorithms* 54(1), 45–57 (2005)
11. Kubat, P., Smith, J.M.: Balancing traffic flows in resilient packet rings. In: Girard, A., et al. (eds.) *Performance evaluation and planning methods for the next generation internet. GERAD 25th Anniversary, Series. 6*, pp. 125–140. Springer, Heidelberg (2005)
12. Cho, K.S., Joo, U.G., Lee, H.S., Kim, B.T., Lee, W.D.: Efficient Load Balancing Algorithms for a Resilient Packet Ring. *ETRI Journal* 27(1), 110–113 (2005)

13. Yuan, J., Zhou, S.: Polynomial Time Solvability Of The Weighted Ring Arc-Loading Problem With Integer Splitting. *Journal of Interconnection Networks* 5(2), 193–200 (2004)
14. Bernardino, A.M., Bernardino, E.M., Sánchez-Pérez, J.M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A.: Solving the non-split weighted ring arc-loading problem in a Resilient Packet Ring using Particle Swarm Optimization. In: *International Conference in Evolutionary Computation* (2009)
15. Pan, Q.-K., Tasgetiren, M.F., Liang, Y.-C.: A discrete differential evolution algorithm for the permutation flowshop scheduling problem. In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pp. 126–133 (2007)
16. Storn, R., Price, K.: *Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*. Technical Report TR-95-012, ICSI (1995)
17. Price, K., Storn, R., Lampinen, J.: *Differential Evolution - A Practical Approach to Global Optimization*. Springer, Berlin (2005)
18. Tasgetiren, M.F., Pan, Q.-K., Liang, Y.-C.: A discrete differential evolution algorithm for the single machine total weighted tardiness problem with sequence dependent setup times. *Computers and Operations Research* 36(6), 1900–1915 (2009)
19. *Differential Evolution Homepage*,
<http://www.icsi.berkeley.edu/~storn/code.html>

A Parallel Genetic Algorithm on a Multi-Processor System-on-Chip

Rubem Euzébio Ferreira¹, Luiza de Macedo Mourelle², and Nadia Nedjah³

¹ Center of Informatics

State University of Rio de Janeiro, Brazil

`frubem@uerj.br`

² Department of Systems Engineering and Computation

Faculty of Engineering

State University of Rio de Janeiro, Brazil

`ldmm@eng.uerj.br`

³ Department of Electronics Engineering and Telecommunications

Faculty of Engineering

State University of Rio de Janeiro, Brazil

`nadia@eng.uerj.br`

Abstract. The aim of the work described in this paper is to investigate migration strategies for the execution of parallel genetic algorithms in a Multi-Processor System-on-Chip (MPSoC). Some multimedia and Internet applications for wireless communications are using genetic algorithms and can benefit of the advantages provided by parallel processing on MPSoCs. In order to run such algorithms, we use a Network-on-Chip platform, which provides the interconnection network required for the communication between processors. Two migration strategies are employed, in order to analyze the speedup and efficiency each one can provide, considering the communication costs they require.

Keywords: Parallel Genetic Algorithm, Multi-Processor System-on-Chip, Network-on-Chip, Migration Topologies.

1 Introduction

The increasing demand of electronic systems, that require more and more processing power, low energy consumption, reduced area and low cost, has lead to the development of more complex embedded systems, also known as System-on-Chip (SoC), in order to run multimedia, Internet and wireless communication applications [9]. These systems can be built of several independent subsystems, that work in parallel and interchange data. When these systems have more than one processor, they are called Multi-Processor System-on-Chip (MPSoC).

Currently, several products, such as cell phones, portable computers, digital televisions and video games, are built using embedded systems. While in embedded systems the communication between Intellectual Property (IP) blocks is basically done through a shared bus, in multiprocessor embedded systems this

kind of interconnection compromises the expected performance [2]. In this case, the communication is best developed using an intrachip network, implemented by a Network-on-Chip (NoC) [6] [5] [1] platform.

Some multimedia and Internet applications for wireless communications are using genetic algorithms and can benefit from the advantages provided by parallel processing on MPSoCs. In this paper, we present a parallel genetic algorithm that runs on Hermes Multi-Processor System (HMPS) architecture and discuss the impact of migration strategies on performance. In Section 2, we describe the HMPS architecture. The parallel genetic algorithm, used in this paper, is presented in Section 3 and some simulation results are introduced in Section 4. Finally, we draw some conclusions and future work in Section 5.

2 Multi-Processor System-on-Chip Platform

Figure 1 shows the Multi-Processor System-on-Chip (MPSoC), called Hermes Multiprocessor System (HMPS) [3]. MPSoC architectures may be represented as a set of processing nodes that communicate via a communication network. Switches compose the network and RISC processors the processing nodes (Plasma). Information exchanged between resources are transferred as messages, which can be split into smaller parts called packages [7]. The switch allows for retransmission of messages from one module to another and decides which path these messages should take. Each switch has a set of bidirectional ports for the interconnection with a resource and the neighboring switches.

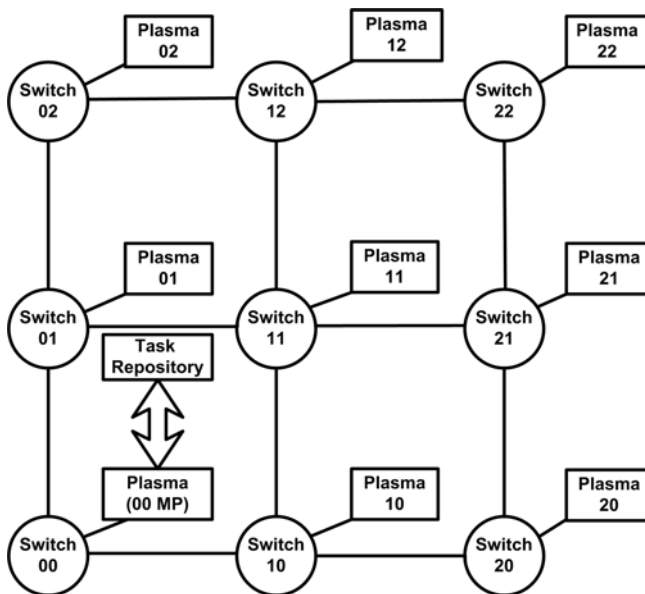


Fig. 1. HMPS architecture, with 9 RISC Plasma processors connected to a 3x3 mesh network

As the total number of tasks composing the target application may exceed the MPSoC memory resources, one processor is dedicated to the management of the system resources (MP - Manager Processor). The MP has access to the task repository, from where tasks are allocated to some processors of the system.

The interconnection network is based on HERMES [4], that implements worm-hole packet switching with a 2D-mesh topology. The HERMES switch employs input buffers, centralized control logic, an internal crossbar and five bi-directional ports. The Local port establishes the communication between the switch and its local IP core. The other ports of the switch are connected to neighboring switches. A centralized round-robin arbitration grants access to incoming packets and a deterministic XY routing algorithm is used to select the output port.

The processor is based on the PLASMA processor [10], a RISC microprocessor. It has a compact instruction set comparable to a MIPS-1, 3 pipeline stages, no cache, no Memory Management Unit (MMU) and no memory protection support in order to keep it as small as possible. A dedicated Direct Memory Access (DMA) unit is also used for speeding up task mapping, but not for data communications. The processor local memory (1024 Kbytes) is divided into four independent pages. Page 0 receives the microkernel and pages 1 to 3 the tasks. Each task can hold 256 Kbytes (0x40000).

The HMPS communication primitives, *WritePipe()* and *ReadPipe()*, essentially abstract communications, so that tasks can communicate with each other without knowing their position on the system, either on the same processor or a remote one. When HMPS starts, only the microkernel is loaded into the local memory. All tasks are stored in the task repository. The manager processor is responsible for reading the object codes from the task repository and transmit them to the other processors. The DMA module is responsible for transferring the object code from the network interfaces to the local memory.

3 Parallel Genetic Algorithm

The Parallel Genetic Algorithm (PGA) is based on the island model, in which serial isolated subpopulations evolve in parallel and each one is controlled by a single processor. This processor periodically sends its best individuals to neighboring subpopulations and receives their best individuals. These individuals are used to substitute the local worst ones. It is obvious that the GA time processing increases with population size. Therefore, small subpopulations tend to converge quickly when isolated.

The PGA is executed by the HMPS platform. Each processor corresponds to an island and its initial subpopulation is randomly generated, evolving independently from the other subpopulations, until the migration operator is activated, as described in Algorithm 1. Premature convergence occurs less in a multi-population GA and can be ignored, when other islands produce better results. Each island can use a different set of GA operators, i.e. crossover and mutation rates, which causes different convergence. Migration of the chromosomes among the islands prevents mono-race populations, which converge

Algorithm 1. PGA

```

1: Initialize the evolutionary parameters
2:  $t \leftarrow 0$ 
3: Initialize a random population  $p(t)$ 
4: Evaluate  $p(t)$  in order to find th best solution
5: while ( $t < NumGenerations$ ) do
6:    $t \leftarrow t + 1$ 
7:   Select  $p(t)$  from  $p(t - 1)$ 
8:   Crossover
9:   Mutation
10:  Evaluate  $p(t)$  in order to find th best solution
11:  if ( $t \bmod MigrationRate = 0$ ) then
12:    Migrate local  $best[p(t)]$  to the next processor
13:    Receive remote  $best[p(t)]$  from the previous processor
14:    Replace  $worst[p(t)]$  by  $best[p(t)]$ 
15:  end if
16: end while

```

prematurely. Periodic migration, which occurs after some generations, prevents a common convergence among the islands.

The PGA requires the definition of some parameters: number of processors, how often the migration will take place, which individuals will migrate and which individuals will be replaced due to migration. The island model introduces a migration operator in order to migrate the best individuals from one subpopulation to another.

3.1 Topology Strategies

In this work, we investigate two topology strategies to migrate individuals from one subpopulation to another: ring and neighborhood. In the ring topology, the best individuals from one subpopulation can only migrate to an adjacent one. As seen in Figure 2, the best individuals from subpopulation 6 can only migrate to subpopulation 1 and the best individuals from subpopulation 1 can only migrate to subpopulation 2. In Algorithm 2, migration is implemented using this kind of strategy. In the neighborhood topology, the best individuals from one subpopulation can migrate to a left and to a right neighbor, as seen in Figure 3. For this kind of strategy, migration is implemented as in Algorithm 3.

Choosing the right time of migration and which individuals should migrate are two critical decisions. Migrations should occur after a time long enough for allowing the development of good characteristics in each subpopulation. Migration is a trigger for evolutionary changes and should occur after a fixed number of generations in each subpopulation. The migrant individuals are usually selected from the best individuals in the origin subpopulation and they replace the worst ones in the destination subpopulation. Since there are no fixed rules that would give good results, intuition is still strongly recommended to fix the migration rate [11].

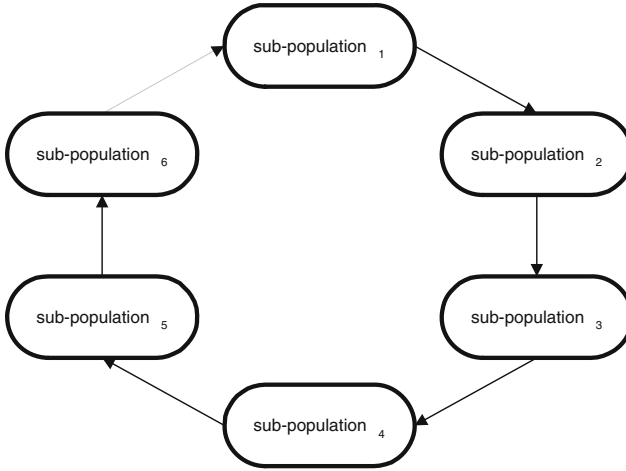


Fig. 2. Ring migration topology

Algorithm 2. Migration function for the ring communication

```

1: local := getprocessid();
2: if local = 0 then
3:   next := 1; previous := number of tasks - 1;
4: end if
5: if local > 0 e local < number of tasks - 1 then
6:   next := local + 1; previous := local - 1;
7: end if
8: if local = number of tasks - 1 then
9:   next := 0; previous := local - 1;
10: end if
11: Send the best individuals to the task, whose identifier is next;
12: Receive the best individuals from the task, whose identifier is previous.
  
```

Algorithm 3. Migration function for the neighborhood communication

```

1: local := getprocessid();
2: if local = 0 then
3:   next := 1; previous := number of tasks - 1;
4: end if
5: if local > 0 e local < number of tasks - 1 then
6:   next := local + 1; previous := local - 1;
7: end if
8: if local = number of tasks - 1 then
9:   next := 0; previous := local - 1;
10: end if
11: Send the best individuals to the task, whose identifier is previous;
12: Send the best individuals to the task, whose identifier is next;
13: Receive the best individuals from the task, whose identifier is previous;
14: Receive the best individuals from the task, whose identifier is next.
  
```

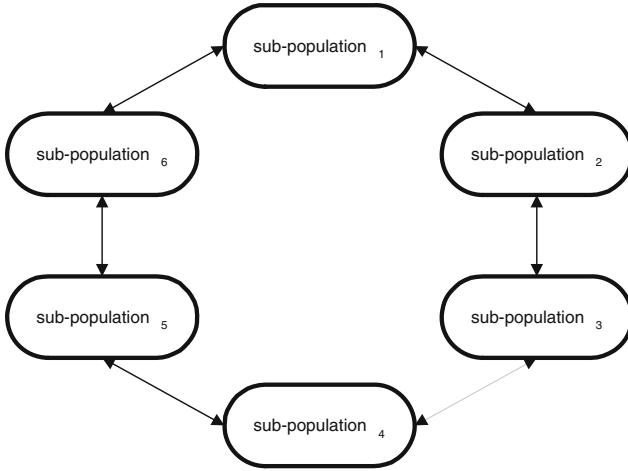


Fig. 3. Neighborhood migration topology

Sending an individual from one subpopulation to another increases the fitness of the destination subpopulation and maintains the population diversity of the other subpopulation. As in the sequential GA, issues of selection pressure and diversity arise. If a subpopulation receives frequently and consistently highly fit individuals, these become predominant in the subpopulation and the GA will focus its search on them at the expense of diversity loose. On the other hand, if random individuals are received, the diversity may be maintained, but the fitness of the subpopulation may not be improved as desired. As migration policy, the best individual is chosen as the migrant, replacing the worst one in the receiving subpopulations. For the migration frequency, an empirical value was adopted based on the number of generations.

4 Simulation Results

The two non-linear functions defined by Equation 1 were used by the PGA for optimization. Function $f_1(x)$ has 14 local maximum e one global maximum in the interval $[-1, 2]$, with an approximate global maximum of 2.83917, at $x = 1.84705$. Function $f_2(x, y)$ has various local minimum and one global minimum in the interval $-3 \leq x \leq 3$ and $-3 \leq y \leq 3$, and an approximate global minimum of -12.92393 , at $x = 2,36470$ and $y = 2.48235$.

$$\begin{aligned} \max_x f_1(x) &= \text{sen}(10\pi x) + 1 \\ \min_{x,y} f_2(x, y) &= \cos(4x) + 3\text{sen}(2y) + (y - 2)^2 - (y + 1) \end{aligned} \quad (1)$$

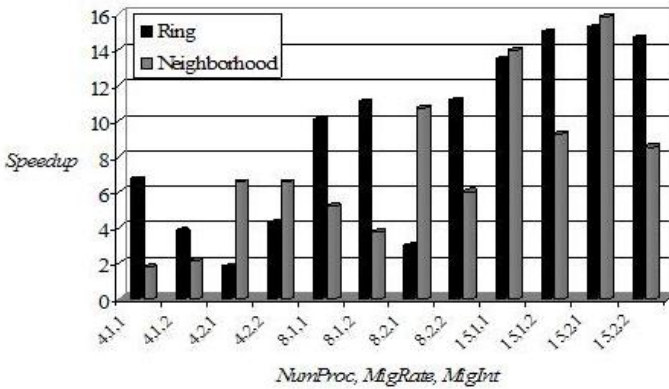
The performance of the PGA can be evaluated based on its speedup and efficiency. Speedup S_p [8] is defined according to Equation 2, where T_1 is the execution time of the sequential version of the genetic algorithm and T_p is the execution time of its parallel version.

$$S_p = \frac{T_1}{T_p} \tag{2}$$

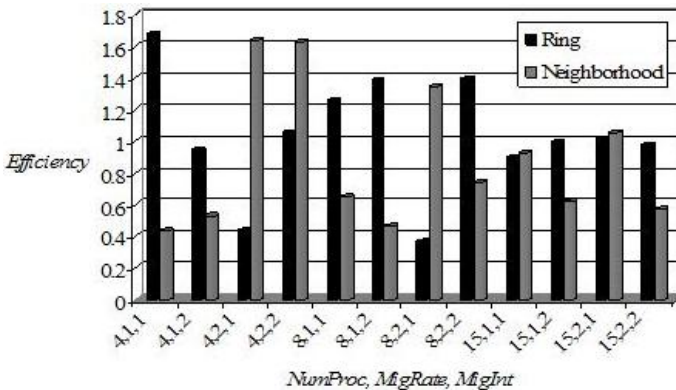
Efficiency E_p [8] is defined according to Equation [3], where $\frac{1}{p} < E_p \leq 1$ and p is the number of processors employed.

$$E_p = \frac{S_p}{p} \tag{3}$$

Based on simulation results for the optimization of $f_1(x)$ and $f_2(x, y)$ using the ring and neighborhood topologies, we obtained the graphics for speedup and efficiency shown in Figure [4] and Figure [5] respectively. The data are presented as triples consisting of the number of slave processors used ($NumProc$), the migration rate ($MigRate$) and the migration interval ($MigInt$).



(a) Speedup of $f_1(x)$



(b) Efficiency of $f_1(x)$

Fig. 4. Impact of the migration rate and migration interval on speedup and efficiency for function $f_1(x)$, considering the used topology

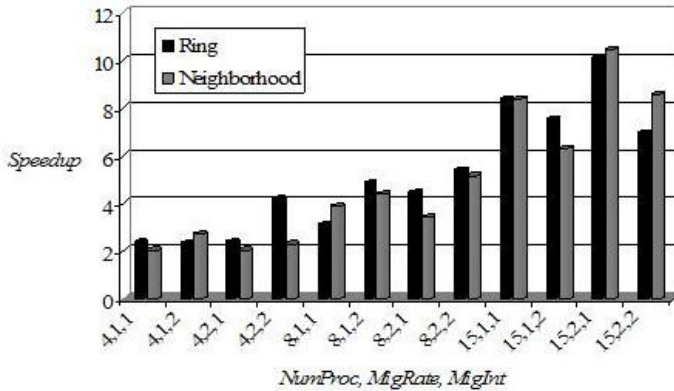
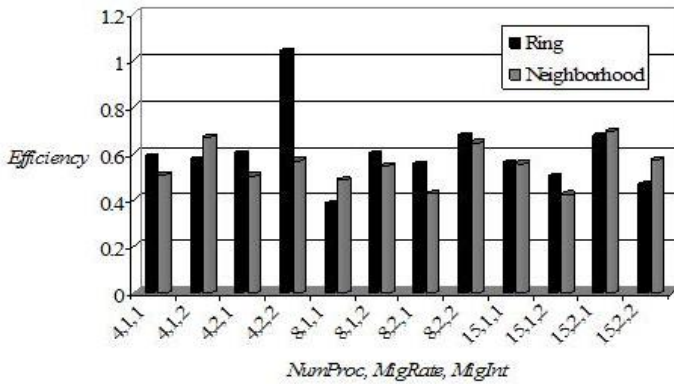
(a) Speedup of $f_2(x, y)$ (b) Efficiency of $f_2(x, y)$

Fig. 5. Impact of the migration rate and migration interval on speedup and efficiency for function $f_2(x, y)$, considering the used topology

5 Conclusions

For the ring topology, the behavior of the two functions shows that, keeping the migration interval constant and varying the migration rate, if the increase in the migration rate resulted in an increase in speedup and efficiency, the fitness of the individuals, received by one or more populations during the migration phase, accelerated the evolutionary process, decreasing the convergence time. On the other hand, if the increase in the migration rate resulted in the decrease of speedup and efficiency, then we can say that the fitness of these individuals did not influence enough the evolutionary process of the populations that received them. In this case, the convergence time increases.

In the future, we intend to investigate the impact of other migration strategies on the performance of the parallel Network-on-chip based implementation of genetic algorithms. One of these topologies is broadcasting, which allows

each processor to send the best solution found so far to all the other processors in the network. We will assess the impact of heavy message send/receive workload on the overall system performance.

References

1. Ivanov, A., De Micheli, G.: The network-on-chip paradigm in practice and research. *IEEE Design and Test of Computers* 1(1), 399–403 (2005)
2. Mello, A.M.: Arquitetura multiprocessada em SoCs: estudo de diferentes topologias de conexão (June 2003) [in Portuguese]
3. Woszezenki, C.: Alocação de tarefas e comunicação entre tarefas em mpsocs. M.Sc., Faculdade de Informática, PUCRS, Porto Alegre, RS, Brazil (June 2007) [in Portuguese]
4. Moraes, F., Calazans, N., Mello, A., Möller, L., Ost, L.: Hermes: an infrastructure for low area overhead packet-switching networks on chip. *Integration, the VLSI Journal* 38(1), 69–93 (2004)
5. Öberg, J., Jantsch, A., Tenhunen, H.: Special issue on networks on chip. *Journal of Systems Architecture* 1(1), 61–63 (2004)
6. Benini, L., De Micheli, G.: Networks on chips: a new soc paradigm. *IEEE Computer* 1(1), 70–78 (2002)
7. Benini, L., Ye, T.T., De Micheli, G.: Packetized on-chip interconnect communication analysis for MPSoC. In: *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE 2003)*, pp. 344–349. IEEE Press, Los Alamitos (2003)
8. Chiwiacowsky, L.D., de Velho, H.F.C., Preto, A.J., Stephany, S.: Identifying initial conduction in heat conduction transfer by a genetic algorithm: a parallel approach 28, 180–195 (April 1980)
9. Ruiz, P.M., Antonio: Using genetic algorithms to optimize the behavior of adaptive multimedia applications in wireless and mobile scenarios. In: *IEEE Wireless Communications and Networking Conference (WCNC 2003)*, pp. 2064–2068. IEEE Press, Los Alamitos (2003)
10. Rhoads, S.: Plasma microprocessor (2009), <http://www.opencores.org>
11. Hue, X.: Genetic algorithms for optimization – background and applications. Technical report. Edinburgh Parallel Computer Centre, The University of Edinburgh (1997)

The Influence of Using Design Patterns on the Process of Implementing Genetic Algorithms

Urszula Markowska-Kaczmar and Filip Krygowski

Wroclaw University of Technology, Insitute of Informatics,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
urszula.markowska-kaczmar@pwr.wroc.pl

Abstract. The design of genetic algorithm is made on the basis of trial by error method, mainly. The aim of the research performed in this work was to examine the effects of using software design patterns in genetic algorithm implementation on the process of modifying the algorithm. Additionally, specific patterns were evaluated from the point of view of their contribution to reducing the difficulty of modifying a system.

Keywords: genetic algorithm, design patterns, software engineering.

1 Introduction

Genetic algorithms are a method of searching solutions in optimisation problems. As the range of problems that can be potentially solved using genetic algorithms varies greatly, so do the algorithms themselves. Furthermore, many parameters (like the number of individuals in a population for instance) affect the algorithm's behaviour. The abundance of variations to choose from and parameters to set can make the process of implementing a genetic algorithm quite lengthy. The final, most efficient configuration can be found by trial and error – different variants are implemented and tested, different values of parameters are tried out.

Design patterns have been used in software development for many years. They provide well-tested solutions to problems commonly encountered during software design. Using patterns improves code robustness and readability. For those and other reasons patterns are applied in various kinds of systems, including systems based on genetic algorithms. But so far research focused mostly on using design patterns in genetic algorithms to create an alterable framework. The research focused on applying the patterns to gain certain benefits, not on analysing the benefits themselves. Furthermore, only the final effects were described, not the implementation process or the role of patterns.

The goal of this research was to examine how the process of modifying a genetic algorithm-based system is affected by the use of design patterns. Two tests have been performed: one based on a case study, the other based on comparing two systems. Both tests used a genetic algorithm system, named GATATest, implemented for this research. It solves a non-trivial, real-life problem.

The content of this paper is organized as follows: genetic algorithms, design and implementation issues are described first, then design patterns as well as detailed information on the patterns applicable to genetic algorithms are presented shortly. Next, the evaluation methods of the design patterns influence is described. The summary finishes the paper.

2 Genetic Algorithms

The idea behind this technique is quite simple: the algorithm starts with a population of random solutions to a given problem. The solutions are tested and evaluated. The ones that prove to be the best reproduce with each other, possibly breeding new, better solutions. Occasionally a random modification - mutation, is introduced in a solution to search through areas not covered by the current population. The algorithm starts from a population of individuals, each representing a solution to the given problem, encoded in a chromosome. The algorithm consists of several steps, as shown in Fig. 1. Its first step is evaluating the fitness of the individuals. In this process each solution is tested and assigned a value representing how well it performed with regard to the rest of the population. This is the individual's fitness value and the better the solution, the greater it is. Next the algorithm's termination criteria are tested. If the criteria are met, the algorithm stops and the best individual from the current population is selected as the result. In the opposite case, a new population of individuals is generated. The process of creating a new population starts with selecting individuals for reproduction. The selection is performed based on the fitness calculated. Better solutions have a higher chance of being selected and can be selected multiple times while inferior solutions might not be selected at all. After proper individuals have been selected, they are joined in pairs. This is most commonly done randomly. The pairs then produce offspring - each pair produces two new individuals. The chromosomes of the offspring are determined by means of crossing over. In the simplest algorithm - one-point crossover - the chromosomes of the parents are cut in two parts (of possibly different length) at the same, randomly selected point, the parts are exchanged and form two new chromosomes.

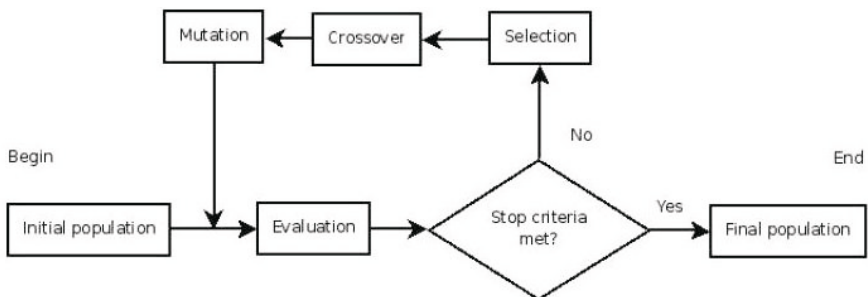


Fig. 1. The basic genetic algorithm

Mutation means that each gene on a chromosome has a (usually small) probability of being changed to a different value. After all mutations have been performed, the new population completely replaces the old one and the algorithm starts from the beginning.

Despite the basic algorithm being constructed using simple ideas and mechanisms, its implementation can be quite complex. There is a great abundance of possible modifications to the basic genetic algorithm. This covers not only the various versions of selection and genetic operations but also the advanced structures of the chromosome and additional extensions. The need of modeling the solution and choosing an appropriate encoding only adds to the complexity of choosing the most efficient options for a single implementation. Researchers conducted many studies on which options to choose and how to best represent the solution for specific kinds of problems. Such research can be very useful in designing and implementing a genetic algorithm. However, in order to make use of such information, the type of problem being solved has first to be identified. Searching for the most suitable selection strategy, changing the representation of the solution or introducing more complicated concepts might require a lot of modifications. A practical approach to this kind of situation would be improving the algorithm's design. If the system itself allowed for easier modifications, the cost of implementing changes to the algorithm would be much lower.

3 Design Patterns

Design patterns in software engineering are general solutions to well-known object design problems. They were thoroughly tested and have proven themselves to be effective. Yet despite the power that lies in software design patterns the solutions they provide are quite simple and are considered to be elegant among software designers.

The literature study performed as part of this research has indicated some patterns that might be useful in such a system. We can mention here the *abstract factory* design pattern, which is responsible for creating objects. It provides the user with an interface for instantiating objects of a family of classes. Whenever an object of such a class is to be created, a method of the appropriate factory class is called to create the object. The *abstract factory* can be used in genetic algorithms to ensure compatibility between different parts of the algorithm. An example would be a factory responsible for creating chromosomes and encapsulated algorithms for crossover and mutation [5].

The goal of the *bridge* design pattern is to separate the implementation from its abstraction. Normally, when the implementation and abstraction are defined by an interface or abstract class and a concrete implementation, they are tied together. The separation provided by the bridge pattern allows either the implementation or the abstraction to be modified without affecting the other.

Similarly to the *abstract factory* pattern, the *builder* pattern is responsible for creating objects. The goal, however, is quite different – builders are used for creating objects of a single class that has a complex structure. The builder's

interface reflects the parts of the object or steps of the creation process. Concrete builders correspond to different representations of the object.

The *decorator* pattern allows to add functionality to existing objects without changing their structure. Furthermore, the functionality is added for a single object, not entire classes. This is done by wrapping an object with a decorator objects that contains the added functionality. The decorators operations are used to access its functions.

The *strategy design* pattern is used to express algorithms as interchangeable objects. Algorithms are encapsulated in classes and instantiated as objects. This way the behaviour of the context utilising a strategy can be changed by using a different strategy object.

The *template method* pattern defines an outline of an algorithm which is implemented in a method. Its main steps are defined as separate methods. Sub-classes of the class containing the algorithm may change the implementation of particular steps, but the overall structure remains the same.

The *visitor* pattern defines operations that are to be performed on objects belonging to an object structure. The implementation of a new operation would require to modify each object class to add the new behaviour. As these classes might be quite numerous, this operation might require great effort. Instead, the visitor pattern encapsulates the operation in a visitor class that visits the object structure.

4 Evaluation Methods

Rationale suggests that using design patterns in the design of a system based on a genetic algorithm should improve the systems flexibility and allow for easier modifications. Applications of patterns to such systems have been described in literature [5], [6], [9], but the actual influence of patterns on the development of the system has not been evaluated. This section describes the methods of verifying the usefulness of design patterns. As the ease of modifying a system is not directly measurable, the tests are based on subjective software developers opinions and measuring software complexity on the basis of Halstead's metrics.

In order to observe the effects of using design patterns in practice a system named GATATest, was designed and constructed. The most of the patterns described in previous section was applied. After the system was implemented, various changes to the algorithm were introduced to test the ease of modification. The range of changes included:

- implementing different selection and crossover methods,
- changing the termination criteria,
- introducing different chromosome representations,
- modifying the evaluation method.

In order to more objectively evaluate the program in this research, Halstead's metrics are used to compare genetic algorithm designs with and without the use of patterns. They treat the program as a collection of tokens that can either be

an operand or an operator [1]. They are measures of a program's complexity. They are calculated using the JHawk tool; following the method described in [8]. These metrics indicate which design is more complex. In general, the more complex a design is, the more difficult it is to modify the code. Therefore, based on the values of the metrics, it can be deduced whether the patterns used improve the process of modifying the algorithm.

Only the *length*, *volume* and *effort* measures are used for further computations. The *length* is defined as the total number of tokens used; it can be calculated by the formula:

$$N = N_1 + N_2, \quad (1)$$

where N_1 is the total number of operators, N_2 is the total number of operands. The metrics *volume* measures the size of the implementation and resembles "the number of mental comparisons required to generate a program" [4].

$$V = N \log_2 n, \quad (2)$$

where n is the sum of n_1 (the number of distinct operators) and n_2 (the number of distinct operands).

V^* is the volume of the minimal implementation of the algorithm and it is calculated as: $V^* = (2 + n_2^*) \log_2 (2 + n_2^*)$. Here n_2^* is the number of potential operands (it represents a minimal number of operands that can represent a computer program and are specific for every programming language).

The last metrics the *effort* measures the amount of mental activity required to implement the algorithm as the measured program:

$$E = \frac{V}{L} \quad (3)$$

where L is level metrics which measures how well a program is written and formally can be expressed as $L = V^*/V$. The measures for a class are sums of the values calculated for individual methods. Similarly, calculating the metrics for a package requires summing the values computed for individual classes. This method can be extended for the whole system by summing the metrics of all packages.

5 The GATATest System

This section describes the GATATest system used in both evaluation methods. It is based on the genetic algorithm which goal is to evaluate sets of market indicators in order to find the one set that allows for the most precise predictions.

For the purpose of the system 3 market indicators from technical analysis were implemented: *simple moving average*, *weighted moving average* and *exponential moving average* [2]. These indicators can be calculated taking into account various numbers of past values. In the system each of the averages uses 15, 30 and 45 last values, which results in 9 different indicators. The indicators are kept in a list structure in the system. The chromosome defines a combination of the indicators in the list. It has the form of a boolean array. A boolean value on the

i -th position in the chromosome codes whether the i -th indicator is part of this solution. Evaluation is based on testing the predictions generated by a set of indicators on real data. The data consists of the values of WIG20 futures contracts on the Warsaw Stock Exchange between 17-Nov-2000 and 28-Apr-2008. The evaluation of a specific chromosome is performed in 2 stages: calculating the objective function and transforming the result to a fitness value which is expressed in number of points reflecting the precision of prediction.

Other important elements of the algorithm include mechanisms of creating the initial population, generating a new population (selection, crossover, mutation) and the termination criteria. In the initial algorithm, these elements were based on the fundamental version of a genetic algorithm [3]. Details can be presented as follows: initialisation – the initial population is created randomly and consists of 50 individuals; selection – the roulette wheel algorithm was used. This algorithm selects parents randomly. The probability of selecting an individual is equal to the individual’s fitness value divided by the sum of the fitness of all individuals. The crossover operation was implemented as one-point crossover. Bitwise mutation [3] is applied (every bit on the chromosome has the probability of 0.1% to be changed to the opposite value). The initial algorithm has only one termination criterion – it will stop after 100 generations have elapsed.

Fig. 2 presents the UML model of the system. It contains only the classes forming the structure of the algorithm. Classes responsible for the problem domain technical analysis – are not included. Furthermore, only the abstract classes are presented for clarity. The *Algorithm* class is the interface for the whole algorithm. It has only one method that runs the algorithm. It is also responsible for presenting the result. The main engine of the system is located in the *Population* class. It coordinates all processes regarding the creation of the initial and subsequent generations, verifying termination criteria etc. The *Individual* class represents an individual in the algorithm. It contains a chromosome, which

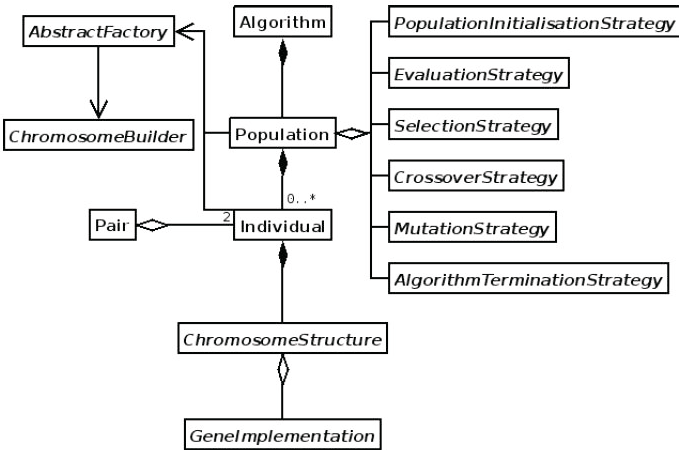


Fig. 2. The UML model of the GATATest system

can be retrieved for processing. It can also be assigned a value – the effect of its evaluation. The function of the *Pair* class is to bind two *Individual* objects together. It is used in the selection and crossover processes to create pairs of individuals for reproduction. The classes *TAIndicator* and *DataElement* are used in several methods, though they were not placed on the model. They represent an abstract technical analysis indicator and a portion of data prepared for analysis, respectively. Other classes in the model are parts of design patterns. The system's design makes extensive use of the *strategy* pattern. The *Population* class forms the general structure of the algorithms and defines its steps (initialisation, evaluation, selection, crossover, mutation, termination criteria test). All these specific steps are implemented as *strategies*. The *Population* class aggregates the strategies and uses them to perform the stages of the algorithm.

The *bridge* pattern is used to separate the structure of the chromosome and the form of the gene. The *ChromosomeStructure* class defines the implementation of the chromosome (array, list, tree). The *GeneImplementation* class specifies the implementation of the gene (boolean, integer, object) and how many genes code a single parameter. It is also responsible for creating and mutating genes. The crossover operation is performed on the structure created by the abstraction, thus the *ChromosomeStructure* and *CrossoverStrategy* implementations need to be compatible. The *MutationStrategy* also needs to be compatible with the abstraction, as it iterates through the genes. The *GeneImplementation* class implements the method of mutating a gene, so it is independent of the mutation operation.

The *builder* pattern creates objects representing chromosomes. The builder is responsible for matching a *ChromosomeStructure* with a *GeneImplementation* and creating the required genes. The *extendChromosome()* method extends the chromosome by adding genes encoding the use of a specific indicator. The *factory* is responsible for instantiating strategy and chromosome objects. Concrete factories also ensure that the created objects are compatible with each other.

Template methods were used in several classes of the system. Good illustrations are the *EvaluationStrategy* and *SelectionStrategy* classes. In the first class, the *evaluate()* method is a template method. The *EvaluationStrategy* class relies on its subclasses to perform the actual evaluation in the *calculateObjectiveFunction()* method. The result is then transformed to a fitness value.

6 Evaluation Study

The first evaluation is based on our subjective observation of easiness level in introducing changes in GATATest code which aim was to improve the performance of the application in terms of achieved fitness function values. During the case study, a total of 9 changes were introduced. Most of the vital elements of the algorithm were modified; only the operation of mutation and the initialisation of the population were unchanged. The description of changes and results obtained by introducing them are presented in Table 6.1. The initial system generated individuals with a maximum value of 1761 points. The results of the final version of GATATest reached the level of 3130 points (fitness function).

Table 1. The effects of the changes on the results and performance of the algorithm: R_{avg} is the average result (in points), R_{max} is the maximum result from all the runs (in points), T_{max} is the average time of a single run (measured in seconds)

| System version | Change | $R_{avg}[P]$ | $R_{max}[P]$ | $T_{avg}[S]$ |
|-----------------|-------------------------------------------|--------------|--------------|--------------|
| Initial version | none | 1660.0 | 1761 | 28.41 |
| Change 1 | saving the best individual | 1758.0 | 1761 | 27.97 |
| Change 2 | lack of improvement termination criterion | 1745.5 | 1761 | 17.22 |
| Change 3 | introducing trust levels | 1758.3 | 1871 | 124.15 |
| Change 4 | improving the selection method | 1774.9 | 1875 | 109.83 |
| Change 5 | two trust levels per indicator | 2054.9 | 2126 | 62.85 |
| Change 6 | negative trust levels | 2782.6 | 3045 | 75.38 |
| Change 7 | chromosomes encoded using real values | 2872.4 | 3130 | 26.54 |
| Change 8 | adding crossover methods for real values | 2791.2 | 3034 | 28.38 |
| Change 9 | list-structured chromosome | 2717.7 | 3082 | 149.61 |

Most of the changes included creating new elements instantiated by the *AbstractFactory*. For each of these changes a concrete factory was created that instantiated the new elements. The final system has 9 concrete factories that resemble the evolution of the system from its initial form to the final version. *Change 1* introduces elitism. *Change 2* causes creation an alternative termination criterion. *Change 3* introduces trust levels. It is expressed as a real value in the range from 0 to 1 (both inclusive) and is encoded in the chromosome using binary code. It is taken into account in the voting process. *Change 4* represents change of selection strategy in genetic algorithm. It is calculated in relation to the weakest individual in the population - the fitness reflects the difference in the value between the given individual and the weakest one. *Change 5* - extends the *Change 3*. The new gene implementation encodes two trust levels for each indicator: one for its rise predictions, the other for fall predictions. *Change 6* introduces the possibility to generate a trust level of negative value which means that when an indicator predicts one type of trend, it is treated as the opposite prediction with a positive trust level of the same value. *Change 7* - it changes the encoding of the trust levels from binary coded values to real values. *Change 8* relies on introducing intermediate crossover and flat crossover. One of the important differences between these operations and one-point crossover used previously is that they create only one offspring from a pair of parents. *Change 9* introduces the structure of the chromosome. The structure was changed from an array to a constant-sized list.

The conducted case study allowed to examine all the applied patterns. *Abstract factory* was the most often engaged pattern. Whenever a new strategy or builder was created and was to be used, a concrete implementation of the *AbstractFactory* class needed to be created or modified. The advantage of using an *abstract factory* was visible. All of the algorithm's configuration was held in one place. The effects of this feature could be clearly seen in changes 8 and 9, where the strategy classes had to be instantiated to match the chromosome.

Evaluating the *bridge* pattern presents some difficulties, as the abstraction and implementation parts can be defined in various ways. In general, the bridge pattern can be viewed as useful for separating the structure and representation of the chromosome. In sum, this pattern would be better suited for genetic algorithm frameworks, where the exact form of the chromosome varies depending on the problem.

The *builder* pattern was used to configure the bridges representing the chromosome. In the presented design the *builder* pattern was applicable because the building process could be divided into separate phases. Each phase resembled the extension of the chromosome to encode another indicator. Yet not all chromosome representations might allow such a division, so the pattern might not be generally appropriate for every genetic algorithm system. Its usefulness is evaluated as rather low.

Effects of the *strategy* pattern application was lowering the effort of changing the algorithm's operations. The strategies used in the system have simple interfaces and new strategy classes can be added easily. Another effect was reducing the complexity of the system.

Thanks to the *template methods* only the abstract classes needed to be modified to gain additional functionality. The added code affected all subclasses automatically. Moreover, the concrete classes were simpler and contained less methods. Two of the design patterns described in section 3 were not used in the GATATest system at all – the *decorator* pattern and the *visitor* pattern. However, further research would allow to examine these patterns.

The second evaluation of the design patterns was based on metrics calculated with the use of the JHawk tool to compare two systems: one designed using design patterns, and one designed without them. As the first system, the GATATest system was used. The other system was created for the purpose of this comparison and is called *Patternless*. Much of the code in the *Patternless* system was reused from GATATest. Only three of Halstead's metrics were calculated: length, volume and effort. Table 2 presents the values of the metrics and information about the number of classes and methods in each system.

The comparison of the two systems shows that GATATest has more classes and methods than *Patternless*. This is mostly caused by the use of patterns – they enforce the creation of additional classes and levels of abstraction. The code is more scattered across the many classes of the system. Furthermore, additional code is needed to provide communication between the classes. Therefore the length and volume metrics are higher in GATATest than in *Patternless* but the effort value is higher. The code in *Patternless* is accumulated in a lesser number of methods so the number of operators and operands in a single method is higher.

Table 2. Halstead's metrics calculated for the systems

| System | Classes | Methods | Length | Volume | Effort |
|--------------------|---------|---------|--------|---------|----------|
| <i>Patternless</i> | 5 | 28 | 672 | 2856.2 | 35032.85 |
| GATATest | 21 | 69 | 1049 | 3861.07 | 33023.6 |

7 Summary

The aim of the presented research was to check whether the software design patterns allow for a much more flexible design. The pattern that prove to have the greatest influence on the systems flexibility were the *abstract factory* and *strategy*. The latter was most useful when applied to the selection and crossover operations of the algorithm. The *template* method had a positive effect, although not as important as the previously mentioned patterns. The *bridge* and *builder* patterns also improved the flexibility of the system but were redundant – the same features could be provided by the abstract factory and simple inheritance, respectively. The drawback of the case study is that it was performed by the authors only that is why to obtain more objective opinion these tests will be conducted by more developers. The computed Halstead's software metrics show that GATATest has greater length and volume values as compared to the other system. Such a result is not surprising – the use of design pattern required the creation of additional classes and code to provide communication between them. Despite this fact, the effort metrics bore higher values for the *Patternless* system. It can be argued that a system that both has more source code and requires less effort to implement, is also less complex.

References

1. Al Qutaish, R.E., Abran, A.: An Analysis of the Design and Definitions of Halstead's Metrics. In: 15-th International Symposium on Software Measurement (IWSM), pp. 337–352 (2005)
2. Achelis, S.B.: Technical analysis from A to Z. McGraw-Hill, New York (1995)
3. Eiben, A.E., Smith, J.E.: Introduction to evolutionary computing. Natural Computing Series. Springer, Heidelberg (2003)
4. Halstead, M.H.: Elements of software science. North-Holland, Amsterdam (1977)
5. Lenaerts, T., Manderick, B.: Building a Genetic Programming Framework: The Added-Value of Design Patterns. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, pp. 196–208. Springer, Heidelberg (1998)
6. Ruican, C., Udrescu, M., Prodan, L., Vladutiu, M.: A Genetic Algorithm Framework Applied to Quantum Circuit Synthesis. Intelligence (SCI) 129, 419–429 (2008)
7. Shi, Z., Chao, L., Ke-qing, H.: A Software Pattern of the Genetic Algorithm – a Study on Reusable Object Model of Genetic Algorithm, Wuhan University. Journal of Natural Sciences 6(1-2), 209–217 (2001)
8. VirtualMachinery. The Halstead metrics, <http://www.virtualmachinery.com/sidebar2.htm>
9. Wick, M.R., Phillips, A.T.: Comparing the template method and strategy design patterns in a genetic algorithm application. ACM SIGCSE Bulletin 34(4), 76–80 (2002)

Obtaining Significant Relations in L-Fuzzy Contexts^{*}

Cristina Alcalde¹, Ana Burusco², and Ramón Fuentes-González²

¹ Dpt. Matemática Aplicada, Escuela Univ. Politécnica
Univ. del País Vasco, Plaza de Europa, 1
20018 - San Sebastián, Spain
c.alcalde@ehu.es

² Dpt. Automática y Computación, Univ. Pública de Navarra
Campus de Arrosadía
31006 - Pamplona, Spain
{burusco,rfuentes}@unavarra.es

Abstract. We use linguistic variables in order to obtain significant relations in L-Fuzzy contexts (L, X, Y, R) that allow us to extract complete information from the L-Fuzzy context. We analyze, in particular, the case of anomalous values in the relation R of the L-Fuzzy context, proposing a replacing method in the case where they are erroneous.

1 Introduction

In some previous works ([5], [6]), we defined an L-Fuzzy context (L, X, Y, R) , with L a complete lattice, X and Y the sets of objects and attributes respectively and $R \in L^{X \times Y}$ an L-Fuzzy relation between the objects and the attributes, as an extension to the fuzzy case of the Formal contexts of Wille ([13]) when the relation between the objects and the attributes that we want to study takes values in a complete lattice L . In order to work with these L-Fuzzy contexts, we use the derivation operators 1 and 2 defined by: $\forall A \in L^X, B \in L^Y$

$$A_1(y) = \inf_{x \in X} \{I(A(x), R(x, y))\} ,$$

$$B_2(x) = \inf_{y \in Y} \{I(B(y), R(x, y))\} .$$

where I is a fuzzy implication operator defined in (L, \leq) , which is decreasing in its first argument, and where A_1 represents, in a fuzzy way, the attributes related to the objects of A and B_2 the attributes related to the objects of B .

The information of the context is visualized by means of the L-Fuzzy concepts which are pairs $(A, A_1) \in (L^X, L^Y)$ with $A \in \text{fix}(\varphi)$ the set of the fixed points of the operator φ , being this one defined by the derivation operators 1 and 2

* Work partially supported by the Research Group “Intelligent Systems and Energy” of the University of the Basque Country, under Grant GIU 07/45 and by the Research Project of the Government of Navarra (Resolution 2031 of 2008).

mentioned above as $\varphi(A) = (A_1)_2 = A_{12}$. These pairs represent, in a fuzzy way, a set of objects that share some attributes and can be interpreted as follows: We focus on those objects and attributes whose membership degrees stand out from the rest.

The set $\mathcal{L} = \{(A, A_1) / A \in \text{fix}(\varphi)\}$ with the order relation \leq defined as: $\forall(A, A_1), (C, C_1) \in \mathcal{L}, (A, A_1) \leq (C, C_1)$ if $A \leq C$ (or eq. $C_1 \leq A_1$) is a complete lattice that is said to be an L-Fuzzy concept lattice ([5], [6]).

Other extensions of the Formal concept analysis to the Fuzzy area are in [12], [4], [10] and [11].

2 Use of Linguistic Labels in L-Fuzzy Contexts

2.1 Linguistic Variables

We begin by summarizing some well-known definitions of fuzzy logic.

A *fuzzy number* [14] is a normal and convex fuzzy set. There are many kinds of fuzzy numbers, e.g. triangle, trapezoid, S-shaped, bell etc. These fuzzy numbers characterize the linguistic variables that will appear next.

Taking the definition of Zadeh [14]: By a *linguistic variable* we mean a variable whose values are words or sentences instead of numbers and that is characterized by a tuple $(V, T(V), U, G, M)$ where V is the name of the variable, $T(V)$ is the set of linguistic labels or values, U is the Universe of discourse, G is a syntactic rule which generates the values of $T(V)$ and M is the semantic rule which assigns to each linguistic value $t \in T(V)$ its meaning $M(t)$.

This meaning of a linguistic label t is defined by a *compatibility function* $c_t : U \rightarrow [0, 1]$ which assigns its compatibility with U to every t .

We will now consider linguistic variables defined in the Universal set $[0, 1]$ where the meaning of the label $M(t)$ is represented by a symmetrical trapezoidal fuzzy number. Specifically, we will use those represented in Fig. 1 (the values a and b define the interval where $c_t(x) = 1$):

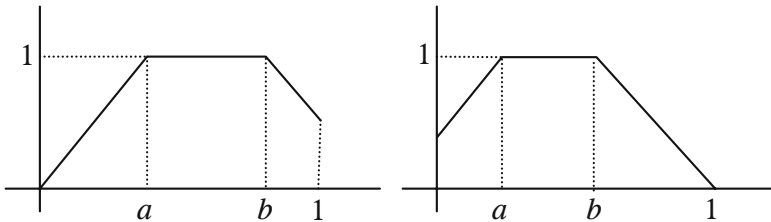


Fig. 1. Fuzzy sets assigned to labels

Observe that these trapezoidal numbers are the restriction to the interval $[0,1]$ of the original ones defined in \mathbb{R} .

Notation. We denote the compatibility of the value $x \in [0, 1]$ with the label t by x_t .

Then, $\forall x \in [0, 1]$:

$$c_t(x) = x_t = \begin{cases} 1 + m(x - a) & \text{if } x \leq a \\ 1 & \text{if } a \leq x \leq b \\ 1 + m(b - x) & \text{if } x \geq b \end{cases}$$

Where $m = \min \left\{ \frac{1}{a}, \frac{1}{1 - b} \right\}$.

These two values, $a, b \in [0, 1]$, are those assigned to label $t \in T(V)$ in its definition.

2.2 Obtaining L-Fuzzy Contexts with Significant Relations

The process of obtaining the relation R of an L-Fuzzy context (with values in a lattice L) that represents the relationship between the set of objects and the other of attributes is not standardized (both where the context takes values in the same rank as in different ranks). In particular, the methods used do not work well when we have a line (row or column) of the relation with very low values since their objects and attributes do not appear in the L-Fuzzy concepts as outstanding elements (and then, as we mentioned in the introduction, will not appear in their interpretation).

Many times, we have concluded that it is better to eliminate those lines (of object or attribute) and to reduce the context of work. Nevertheless, if we eliminate the objects and the attributes, then we will lose information. For example, some of the existing relationships between the objects and the attributes will disappear.

To solve this problem, given a certain group of objects that we want to analyze, we will try to obtain relevant values in each one of the columns of the L-Fuzzy context by means of a linguistic variable that clarifies the different attributes but that, in addition, indicates to us how the original values have been transformed. To do this, for every attribute of the context we will choose its *best linguistic label*. (Analogous, we can interchange the role of the objects and the attributes.)

We are going to use a linguistic variable \mathbf{V} whose set of terms or labels allows us to classify the values of the relation according to its proximity to 0 or 1. It is important that $\forall x \in L, \exists ! t \in T(V)$ such that $x_t = 1$. For this reason, we will use trapezoidal fuzzy labels.

In this way, if we take as a departure point an L-Fuzzy context (L, X, Y, R) , we are going to transform some elements of Y (those with low values) by the linguistic variable \mathbf{V} that will change the context.

Then, for every attribute y_j , we are going to see how we can assign the label: We take the values of the relation corresponding to the attribute y_j , we obtain their maximum M_{y_j} and we choose the only label t verifying that $M_{y_j t} = 1$. That is, $a \leq M_{y_j} \leq b$, where a and b are the values associated with label t in its definition. We will denote this label by t_{y_j} .

Definition 1. For every $y_j \in Y$, the obtained label t_{y_j} is said to be the best linguistic label associated with attribute y_j .

In the same way, we can assign its best linguistic label t_{x_i} to every object x_i .

Then, let $Z \subseteq Y$ be the set of attributes y_j with low values of $R(x_i, y_j)$, $\forall x_i \in X$. We can take the best linguistic label $t_{y_j} \in T(V)$ for every attribute $y_j \in Z$ in order to transform these attributes into others more relevant.

Let (L, X, Y, R) be an L-Fuzzy context and $C = \{(y_j, t_{y_j}), y_j \in Z \subseteq Y, t_{y_j} \in T(V)\}$ that associates linguistic labels to the elements of Z . We are going to give the following definition:

Definition 2. The L-Fuzzy context (L, X, Y^C, R^C) , where $Y^C = (Y \setminus Z) \cup \{y_{j t_{y_j}}, \forall y_j \in Z\}$ and

$$R^C(x_i, y_j) = \begin{cases} R(x_i, y_j) & \text{if } y_j \in Y \\ R(x_i, y_j)_{t_{y_j}} & \text{in other case} \end{cases}$$

is said to be a labeled L-Fuzzy context.

In the same way, we can define the new L-Fuzzy context taking as a departure point the object set.

Next, we will see how this change influences the calculation of the L-Fuzzy concepts associated with the basic points.

Proposition 1. If $A \in L^X$ is a basic point,

$$A(x) = \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{in other case} \end{cases}$$

then, $A_1(y) = R^C(x_i, y) \in L, \forall y \in Y^C$, is the L-Fuzzy concept intension obtained taking A as a departure point. Moreover, the extension verifies that $A_{12}(x_i) = 1$.

Proof. If $A \in L^X$ is a basic point, to calculate the L-Fuzzy concept derived from A , we apply the derivation operator and, using a residuated implication operator (for example, the Lukasiewicz one), we obtain the intension of the concept:

$$A_1(y) = \inf_{x \in X} \{I(A(x), R^C(x, y))\} = R^C(x_i, y), \quad \forall y \in Y^C .$$

Therefore, if $y_{j t_{y_j}} \in Y^C$ is one of the attributes modified by label t_j , the modified values $R^C(x_i, y_{j t_{y_j}})$ can be found in the intension of the corresponding L-Fuzzy concept:

$$A_1(y_{j t_{y_j}}) = R^C(x_i, y_{j t_{y_j}}) \quad \forall x_i \in X, y_{j t_{y_j}} \in Y^C, t_{y_j} \in T(V) .$$

On the other hand, with respect to the intension of the L-Fuzzy concept:

$$A_{12}(x) = \inf_{y \in Y^c} \{I(A_1(y), R^C(x, y))\} = \inf_{y \in Y^c} \{I(R^C(x_i, y), R^C(x, y))\} .$$

Then, we can say that $A_{12}(x_i) = 1$. □

That is, we have some L-Fuzzy concepts where the modified attributes appear outstanding in the new labeled L-Fuzzy context. This is going to allow us to analyze the behavior of these attributes with respect to the different objects of the context.

Example 1. Let (L, X, Y, R) be the L-Fuzzy context represented in Table 1 where the values of column y_2 are quite low.

Table 1. L-Fuzzy context

| R | y_1 | y_2 | y_3 |
|-------|-------|-------|-------|
| x_1 | 0.7 | 0.2 | 1 |
| x_2 | 0 | 0.1 | 0.8 |
| x_3 | 1 | 0.4 | 0.6 |
| x_4 | 0.3 | 0 | 0.9 |

These are the L-Fuzzy concepts derived from the basic points x_1, x_2, x_3 and x_4 using the Lukasiewicz implication operator:

$$\begin{aligned} x_1 &\rightarrow \{(x_1/1, x_2/0.3, x_3/0.6, x_4/0.6), (y_1/0.7, y_2/0.2, y_3/1)\} \\ x_2 &\rightarrow \{(x_1/1, x_2/1, x_3/0.8, x_4/0.9), (y_1/0, y_2/0.1, y_3/0.8)\} \\ x_3 &\rightarrow \{(x_1/0.7, x_2/0, x_3/1, x_4/0.3), (y_1/1, y_2/0.4, y_3/0.6)\} \\ x_4 &\rightarrow \{(x_1/1, x_2/0.7, x_3/0.7, x_4/1), (y_1/0.3, y_2/0, y_3/0.9)\} \end{aligned}$$

As can be seen, the membership degree of y_2 is not one of the highest in any of the L-Fuzzy concepts. Thus, as we mentioned in the introduction, this attribute will not appear in the interpretation of the L-Fuzzy concepts.

We are going to transform the column y_2 into another one with higher values. To do this, we take label t_{y_2} with value 1 for the maximum of the column. In this case, label $t_{y_2} = \textit{medium}$ is the result (with values $a = 0.4$ and $b = 0.6$ in its definition) obtaining the relation of Table 2.

Table 2. New relation R^C

| R^C | y_1 | $y_{2\textit{medium}}$ | y_3 |
|-------|-------|------------------------|-------|
| x_1 | 0.7 | 0.5 | 1 |
| x_2 | 0 | 0.2 | 0.8 |
| x_3 | 1 | 1 | 0.6 |
| x_4 | 0.3 | 0 | 0.9 |

The L-Fuzzy concepts calculated from the basic points in this new L-Fuzzy context (L, X, Y^C, R^C) are:

$$\begin{aligned}
 x_1 &\rightarrow \{(x_1/1, x_2/0.3, x_3/0.6, x_4/0.5), (y_1/0.7, y_{2medium}/0.5, y_3/1)\} \\
 x_2 &\rightarrow \{(x_1/1, x_2/1, x_3/0.8, x_4/0.8), (y_1/0, y_{2medium}/0.2, y_3/0.8)\} \\
 x_3 &\rightarrow \{(x_1/0.5, x_2/0, x_3/1, x_4/0), (y_1/1, y_{2medium}/1, y_3/0.6)\} \\
 x_4 &\rightarrow \{(x_1/1, x_2/0.7, x_3/0.7, x_4/1), (y_1/0.3, y_{2medium}/0, y_3/0.9)\}
 \end{aligned}$$

In this case, from the L-Fuzzy concept assigned to x_3 , we can say that x_3 is mainly associated with a high value of attribute y_1 and a medium value of y_2 .

This best label association process arises initially for those objects (or attributes) with low values due to the problem that has been explained. Nevertheless, it is possible to assign labels to all the objects (or attributes) as has been done in the case of Many-valued contexts [13,9] in the Formal concept theory by means of scales. This will allow us to have a general study of the L-Fuzzy context using the same tool in all cases.

3 L-Fuzzy Contexts with Anomalous Values

3.1 Use of the Labels Associated with Objects and Attributes in L-Fuzzy Contexts with Anomalous Values

The described process of assigning its best label to every object or attribute of the L-Fuzzy context can not be very suitable when we have anomalous values in the context.

We will understand by an anomalous value of an L-Fuzzy context, that which is not similar to the rest of the values of its row, nor of its column. If these anomalous values are the maximum of the corresponding row or column, they are those that determine the label that we associate, even though the rest of the values of the row or column do not fit well with them. On the other hand, only some of those anomalous values will be erroneous and, in the cases which we can verify this, it will be interesting to replace them.

We will follow these steps:

- For every object x_i and attribute y_j , we obtain the maximum M_{x_i} and M_{y_j} of its row and column in order to associate their best labels (t_{x_i} and t_{y_j}) by the process explained in the previous section.
- We substitute every maximum by 0 and then, we calculate the new labels in the resulting relation. In the case of the new labels $t_{x_i}^*$ and $t_{y_j}^*$ change with respect to the old ones, $R(x_i, y_j)$ will be an anomalous value.
- We will analyze (if it is possible) if the anomalous value is erroneous and, in that case, we will replace it.

Example 2. Let (L, X, Y, R) be the L-Fuzzy context represented in Table 3.

And let \mathbf{V} be the linguistic variable which labels $\{very - high, high, medium, low, very - low\}$ are associated to the fuzzy numbers defined by the intervals $[1, 1]$, $[0.7, 0.9]$, $[0.4, 0.6]$, $[0.1, 0.3]$ and $[0, 0]$ respectively (see Fig. 2).

Table 3. Relation of the L-Fuzzy context (L, X, Y, R)

| R | y_1 | y_2 | y_3 | y_4 |
|-------|-------|-------|-------|-------|
| x_1 | 0.4 | 0.5 | 0.4 | 0.7 |
| x_2 | 0.3 | 1 | 0.3 | 0.6 |
| x_3 | 0.5 | 0.3 | 0.5 | 0.8 |
| x_4 | 0.6 | 0.1 | 0.7 | 0.7 |

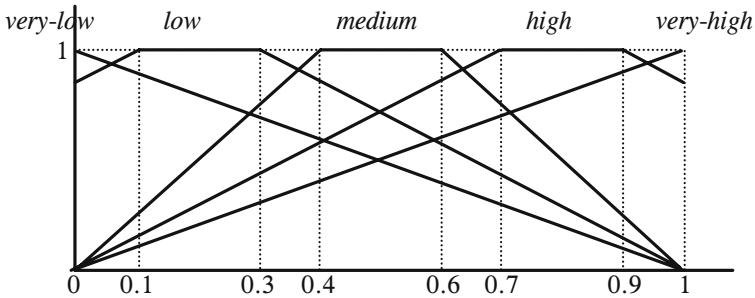


Fig. 2. Labels of the linguistic variable \mathbf{V}

Analyzing the relation R , we can find the value $R(x_2, y_2)$ whose maximum values of its row and column are $M_{x_2} = 1$ and $M_{y_2} = 1$. Thus, $t_{x_2} = \text{very-high}$ and $t_{y_2} = \text{very-high}$ are their *best labels*.

We substitute now these maximum values by 0 and calculate the new *best labels* obtaining $t_{x_2}^* = \text{medium}$ and $t_{y_2}^* = \text{medium}$. As the two labels have changed, we can conclude that the value $R(x_2, y_2)$ is anomalous and we will replace it in the cases where it has been proven to be erroneous.

3.2 Replacement of the Erroneous Values

Once the erroneous values have been detected, following the proposed idea for the case of absent values in the interval-valued L-Fuzzy contexts [13], we will develop the next process to replace these erroneous values:

Let (L, X, Y, R) be the L-Fuzzy context where we have detected that $R(x_i, y_j)$ is erroneous. We will try to replace this erroneous value using linguistic variables and implication between attributes [2] of this type:

If y_k is high (medium, low) then y_j is low (medium, high).

To do this, we take a linguistic variable \mathbf{V} whose set of labels $T(V)$ contains the labels used in the implications. Then, for every attribute $y_j \in Y$ and for every label $t \in T(V)$ we obtain a new attribute y_{j_t} . And we get a new L-Fuzzy context $(L, X, Y \cup \widehat{Y}, \widehat{R})$ where $\widehat{Y} = \{y_{j_t}/y_j \in Y, t \in T(V)\}$ and the relation \widehat{R} is extended $\forall x_i \in X$ to the new attributes in the following way:

$$\widehat{R}(x_i, y_j) = \begin{cases} R(x_i, y_j) & \text{if } y_j \in Y \\ R(x_i, y_j)_t & \text{in other case} \end{cases}$$

In this new context, we analyze if, excluding the object x_i for which we have the erroneous value, it is possible to find any implication of the type $y_{kt_1} \Rightarrow y_{jt_2}$, that is verified with high values of support and confidence [2]. In this case, if an object has the attribute y_{kt_1} to a certain degree, it will also have the attribute y_{jt_2} at the same level. Extending this result to the object x_i the erroneous value can be estimated.

Example 3. We are going to return to the L-Fuzzy context (L, X, Y, R) that we have considered in Example 2, given by Table 3 where we know that the value $R(x_2, y_2)$ is anomalous and we are going to suppose that we have proven that it is also erroneous.

The linguistic variable \mathbf{V} that we are using is given by $\{very - high, high, medium, low, very - low\}$ labels, associated to the fuzzy numbers defined by the intervals $[1, 1]$, $[0.7, 0.9]$, $[0.4, 0.6]$, $[0.1, 0.3]$ and $[0, 0]$ respectively.

We analyze the implications between attributes in the set of objects $X \setminus \{x_2\}$, in order to obtain those that are fulfilled with a high degree of support and confidence. Thus, if we analyze the implication $y_{1low} \Rightarrow y_{2low}$ we have to add to the context these new attributes and extend the relation with two new columns obtained from the compatibility of the initial attributes with the corresponding labels. The relation of the new context is given in Table 4.

Table 4. New relation \widehat{R}

| \widehat{R} | y_1 | y_2 | y_3 | y_4 | y_{1low} | y_{2low} |
|---------------|-------|-------|-------|-------|------------|------------|
| x_1 | 0.4 | 0.5 | 0.4 | 0.7 | 0.9 | 0.7 |
| x_2 | 0.3 | 1 | 0.3 | 0.6 | 1 | 0 |
| x_3 | 0.5 | 0.3 | 0.5 | 0.8 | 0.7 | 1 |
| x_4 | 0.6 | 0.1 | 0.7 | 0.7 | 0.6 | 1 |

Considering only the objects of the set $X \setminus \{x_2\}$ in the new relation \widehat{R} , we obtain high values of support and confidence for the implication between attributes $y_{1low} \Rightarrow y_{2low}$:

$$\begin{aligned} \text{supp}(y_{1low} \Rightarrow y_{2low}) &= \frac{2}{3} = 0.67 \text{ ,} \\ \text{conf}(y_{1low} \Rightarrow y_{2low}) &= \frac{2}{2.2} = 0.91 \text{ .} \end{aligned}$$

We can conclude that in a high percentage of cases (67%), we have the attributes y_{1low} and y_{2low} . Furthermore, in a 91% of cases, where the attribute y_{1low} appears, the membership degree of attribute y_{2low} is at least the same. As this percentage is high, we can expand the result and suppose that it is verified also in the case of object x_2 , and then:

$$\widehat{R}(x_2, y_{2low}) \geq \widehat{R}(x_2, y_{1low}) = 1 .$$

Thus, we have:

$$R(x_2, y_2)_{low} = \widehat{R}(x_2, y_{2low}) = 1 .$$

Then, taking into account the definition of label *low*, we can conclude that the value $R(x_2, y_2) \in [0.1, 0.3]$ and we will choose the medium point of this interval (rounding the values when it is necessary in order to obtain elements of the lattice L), to replace the erroneous value. That is, we will replace the erroneous value by $R(x_2, y_2) = 0.2$.

4 Conclusions

We have shown in this work how the linguistic variables can be very useful to obtain significant relations in the L-Fuzzy contexts and to locate anomalous values. This is not the only possible application. We will see in future works how they can also be used to represent those initial situations that we intend to analyze by means of the study of the derived L-Fuzzy concepts, or to analyze more thoroughly the anomalous values of an L-Fuzzy context.

References

1. Alcalde, C., Burusco, A., Fuentes-González, R.: Treatment of the incomplete information in L-Fuzzy contexts. In: EUSFLAT-LFA 2005, Barcelona, September 2005, pp. 518–523 (2005)
2. Alcalde, C., Burusco, A., Fuentes-González, R.: Implications between attributes in an L-Fuzzy context based on association rules. In: IPMU 2006, Paris, July 2006, pp. 1403–1410 (2006)
3. Alcalde, C., Burusco, A., Fuentes-González, R., Zubia, I.: Treatment of L-Fuzzy contexts with absent values. *Information Sciences* 179(1-2), 1–15 (2009)
4. Bělohávek, R.: Fuzzy Galois connections and fuzzy concept lattices: from binary relations to conceptual structures. In: Novak, V., Perfilova, I. (eds.) *Discovering the World with Fuzzy Logic*, pp. 462–494. Physica-Verlag (2000)
5. Burusco, A., Fuentes-González, R.: The Study of the L-Fuzzy Concept Lattice. *Mathware and Soft Computing* 1(3), 209–218 (1994)
6. Burusco, A., Fuentes-González, R.: Construction of the L-Fuzzy Concept Lattice. *Fuzzy Sets and Systems* 97(1), 109–114 (1998)
7. Burusco, A., Fuentes-González, R.: Fuzzy extensions of the Formal Concept Analysis. In: *International Conference on knowledge, logic and information*, Darmstadt (February 1998)
8. Cousot, P., Cousot, R.: Constructive versions of Tarski's fixed point theorems. *Pacific J. Maths* 82, 43–57 (1979)
9. Ganter, B., Stahl, J., Wille, R.: Conceptual measurement and many-valued contexts. In: Gaul, W., Schader, M. (eds.) *Classification as a tool of research*, pp. 169–176. North Holland, Amsterdam (1986)

10. Medina, J., Ojeda-Aciego, M., Ruiz-Calviño, J.: On multi-adjoint concept lattices: denition and representation theorem. In: Kuznetsov, S.O., Schmidt, S. (eds.) ICFCA 2007. LNCS (LNAI), vol. 4390, pp. 197–209. Springer, Heidelberg (2007)
11. Medina, J., Ojeda-Aciego, M., Ruiz-Calviño, J.: Formal concept analysis via multi-adjoint concept lattices. *Fuzzy Sets and Systems* 160(2), 130–144 (2009)
12. Pollandt, S.: *Fuzzy Begriffe: Formale Begriffsanalyse unscharfer Daten*. Springer, Heidelberg (1997)
13. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) *Ordered Sets*, pp. 445–470. Reidel, Dordrecht-Boston (1982)
14. Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning-I. *Information Sciences* 8, 199–249 (1975)

Knowledge Extraction Based on Fuzzy Unsupervised Decision Tree: Application to an Emergency Call Center

Francisco Barrientos^{1,*} and Gregorio Sainz^{2,1}

¹ CARTIF Centro Tecnológico, Parque Tecnológico de Boecillo 205,
47151 Valladolid, Spain

² University of Valladolid, Systems Engineering and Control Department
School of Industrial Engineering, 47011 Valladolid, Spain

Abstract. This paper describes the application of a fuzzy version of Unsupervised Decision Tree (UDT) to the problem of an emergency call center. The goal is to obtain a decision support system that helps in the resource planning, reaching a trade-off between efficiency and quality of service. To reach this objective, the different types of days have been characterized based on variables that permits available resources assignment in an easy and understandable way. In order to deal with availability of expert knowledge on the problem, an unsupervised methodology had to be used, so fuzzy UDT is a solution merging decision trees and clustering, providing the performance of both viewpoints. Quality indexes give criteria for the selection of a reasonable solution to the complexity, as well as interpretability of the trees and the quality of generated clusters, and also the type of days and the performance from the resources point of view.

Keywords: Unsupervised decision trees, fuzzy clustering, interpretability, emergency call center, decision support systems.

1 Introduction

The main problem in the design and management of any call center is to achieve an adequate trade-off between Quality of Service (QoS) and efficiency. A medium-large sized center can manage thousands of calls per day, and each one must be answered in very few seconds [1].

The management of a call center needs to know the approximate workload to be resolved in a given period, and only then is it possible to estimate the availability of resources to respond to this demand in accordance with the security and quality protocols established in each case. In this way, the center can reach the desired quality standards as well as the user's satisfaction [2].

* This work has been partially supported by the Regional Government of Castilla y León through the Agencia de Protección Civil y Consumo.

The scope of this study is focused on emergency calls, taking into account the maximum availability of the service in order to avoid missing calls. Here, economic factors are not considered.

In this context, the availability of some decision taking system to support the call center managers, giving them help on the out of range service operation, which means the service is operating outside the expected parameters/standards, so as to mobilize the resources that may be needed in time. This work proposes an approach based on a fuzzy version of Unsupervised Decision Trees to reach a linguistic description of day categories involving the workload and resources required for each day category in the call center.

This paper is organized as follows: Section 2 gives a short description of the problem of resource management for an emergency call center and surveys the theoretical basis that supports the model. Section 3 explains the proposal of this work; then the methodology and the experimentation carried out and the analysis of results are given. Finally, the main conclusions and further research are outlined.

2 Emergency Call Center Problem

Call centers have been modeled by queuing theory usually, using the paradigm of producer/consumer. Another possible approach is to model the received calls as a time series and use time series forecasting to estimate the required resources to render the service [2].

In this case, another approach is suggested using a fuzzy version of the Unsupervised Decision Trees (UDT) [3] to characterize the days based on several variables, some intrinsic and others extrinsic to the system (Table 1), and which may be relevant for the call center workload. Based on historical data, estimations and forecasts of these variables, the system will characterize the next few days, and the associated workload, which will give support on making decisions concerning the availability and planning of resources to reach the service standards. This solution appears to be a good alternative due to the difficulty of getting expert knowledge on this focus of the problem. The UDT, unlike other unsupervised learning methods, does not hide the information on the characterization of each class and the involved attributes on the class assigned decision making, so this performance permits to obtain a linguistic description of the solution in accordance with the domain.

2.1 Decision Trees

The fact that two of the “*Top Ten Algorithms in Data Mining*” [4] are tree-based algorithms demonstrates the wide popularity of these methods in the field of data mining. Decision trees are perhaps one of the most widely used paradigms in the world of machine learning, because of their characteristics [5]. A key factor that has influenced its spread is the fact that there are different free implementations available.

Table 1. Input variables summary

| Num | Name | Type | Min | Max | Mean | Std. Dev. |
|-----|-----------------------|-----------|---------|---------|---------|-----------|
| 1 | Non_Inc | intrinsic | 64.90 | 92.70 | 83.38 | 4.33 |
| 2 | Info | intrinsic | 0.30 | 16.30 | 1.89 | 1.35 |
| 3 | Incidents | intrinsic | 6.00 | 31.20 | 14.72 | 4.34 |
| 4 | Sanitary | intrinsic | 33.70 | 75.20 | 57.03 | 5.94 |
| 5 | Security | intrinsic | 18.60 | 57.10 | 36.30 | 5.65 |
| 6 | Search & Rescue (SAR) | intrinsic | 0.30 | 24.50 | 5.04 | 3.69 |
| 7 | Basic_Serv. | intrinsic | 0.00 | 9.90 | 1.64 | 1.35 |
| 8 | Call_Midnight | intrinsic | 8.00 | 495.00 | 163.33 | 91.83 |
| 9 | Call_Dawn | intrinsic | 0.00 | 214.00 | 33.44 | 21.89 |
| 10 | Call_Morning | intrinsic | 92.00 | 722.00 | 245.76 | 98.60 |
| 11 | Call_Afternoon | intrinsic | 109.00 | 835.00 | 360.07 | 132.07 |
| 12 | Call_Evening | intrinsic | 52.00 | 770.00 | 237.51 | 118.68 |
| 13 | Total_Calls | intrinsic | 1658.00 | 9504.00 | 4788.09 | 1608.71 |
| 14 | T_Avg | extrinsic | -3.37 | 28.73 | 11.79 | 7.68 |
| 15 | T_Max | extrinsic | 1.30 | 38.90 | 19.42 | 9.01 |
| 16 | T_Min | extrinsic | -11.90 | 17.90 | 3.92 | 6.36 |
| 17 | Rain | extrinsic | 0.00 | 192.20 | 6.10 | 15.07 |
| 18 | Snow | extrinsic | 0.00 | 6.00 | 0.17 | 0.67 |
| 19 | Fog | extrinsic | 0.00 | 6.00 | 0.54 | 1.12 |
| 20 | Day_of_Week | extrinsic | 1.00 | 7.00 | 4.00 | 2.01 |
| 21 | Month | extrinsic | 1.00 | 12.00 | 6.01 | 3.55 |

2.2 Unsupervised Decision Trees

Recently, research on Unsupervised versions of Decision Trees have been done [3], with hybrid solutions between trees and clustering [5] techniques. This approach expects to combine the advantages of both methods: the classification of the data without prior information, based only on the values of the considered attributes, and the easy interpretation of the results, since each leaf node represents a cluster, and the path from the root to each leaf node represents a classification rule in the *if-then-else* form based on linguistic premises.

Several criteria must be considered for fuzzy UDT: Inhomogeneity Threshold (IT) is related to the content of information in a data set. It is minimal when data are entirely homogeneous, and information content increases when data inhomogeneity increases. This threshold determines whether the node must be segmented. Segment Size Threshold (SST) is the minimum number of samples in a leaf node. If a node exceeds the inhomogeneity threshold it must be divided into segments of at least SST size, and each segment will be a new leaf node.

2.3 Fuzzy Clustering

The goal of clustering is the classification of objects according to similarities between them and the organization of data into groups. The word “similar” must

be understood as mathematical similarity, measured in a well-defined meaning, e.g. the Euclidean distance between two elements.

Since there are some similarities between the techniques of clustering and decision trees, some quality indexes originally defined for fuzzy clustering domain, could be used to check the quality of the results obtained with unsupervised fuzzy trees, complementing those of the trees themselves.

2.4 Quality Indexes

In UDT we cannot use data sets to estimate the classification error. Therefore, other quality indexes must be used to determine the goodness of the results. Here, tree performance (*T-measure*, *IQN-T*, *IC*) [5] [6] and partition/clustering quality measures (*SC*, *S*, *XB*) [8] [9] have been considered, as well as the combination of both measure types to reach a solution with a good trade-off.

T-measure. $T \in [0, 1)$ evaluates the efficiency of a decision tree [5], where a value of 0 is undesirable and a value close to 1 signifies a good decision tree.

$$T = \frac{2n - \sum_{i=1}^{N_{Inodes}} w_i d_i}{2n - 1}, \tag{1}$$

where

$$w_i = \begin{cases} \frac{N_i}{N} & \text{for a resolved leaf node} \\ \frac{2N_i}{N} & \text{otherwise} . \end{cases}$$

IQN-T. The quality of a node Ω (called *IQN* for Impurity Quality Node) is defined [6] as a combination between its purity and its depth.

$$IQN_T(\Omega) = (1 - \varphi(\Omega))f(depth_T(\Omega)) , \tag{2}$$

where $\varphi(\Omega)$ is an impurity measure normalized between $[0, 1]$. In this case, impurity has been defined as the average distance of all the data belonging to a leaf node to its centre of mass and $f(x) = x$. This gives us an idea of how compact the classes defined by the leaf nodes are. Unlike previous measure, this one takes precedence over more expanded trees, since nodes are better suited to the available training examples, even with the risk of falling into overfitting [7].

IC. This index is based on the combination of (1) and (2), since each one promotes a contradictory performance, then a new tree quality index, called IC, is obtained.

$$IC = (1 - d(T, IQN_T)) , \tag{3}$$

where $d(T, IQN_T)$ is the distance between the previous indexes, seeking a trade-off between them in order to have a tree neither too compact nor too expanded.

SC. Partition Index is the ratio of the sum of compactness and separation of the clusters [8]. It is useful when comparing different partitions having an equal number of clusters. A lower value of *SC* indicates a better partition.

$$SC(c) = \sum_{i=1}^c \frac{\sum_{j=1}^N (\mu_{ij})^m \|x_j - v_i\|^2}{N_i \sum_{k=1}^c \|v_k - v_i\|^2}. \tag{4}$$

S. Separation Index, as opposed to the partition index (SC), uses a minimum-distance separation for partition validity [8].

$$S(c) = \frac{\sum_{i=1}^c \sum_{j=1}^N (\mu_{ij})^2 \|x_j - v_i\|^2}{N \min_{i,k} \|v_k - v_i\|^2}. \tag{5}$$

XB. Xie and Beni’s Index aims to quantify the ratio of the total variation within clusters and the separation of clusters [9]. The optimal number of clusters should minimize the value of the index.

$$XB(c) = \frac{\sum_{i=1}^c \sum_{j=1}^N (\mu_{ij})^m \|x_j - v_i\|^2}{N \min_{i,j} \|x_j - v_i\|^2}. \tag{6}$$

3 Methodology

The first step is the acquisition of raw data, whatever its origin or format. Then, it is necessary to do a data pre-processing, to standardize formats, normalize value ranges, missing data recovery, and so on. Once data are ready, an initial feature selection is performed, simplifying the problem by eliminating those variables that do not provide enough information or are redundant, etc. Next, the UDT is carried out using the previously selected variables. The tree performs a second selection, determining what attributes are most relevant to the problem and in what order. Finally, we analyze the results according to previously defined indexes.

3.1 Data Gathering and Pre-processing

The initially available data to build the tree are: the daily number of calls and their classification from January 2005 to April 2007, the calendar with the events that happen each day, and data about the weather in this time period.

The relevance of the historical data is not known "a priori", so in this case a heuristic limit was set: 28 days (4 weeks prior to the actual day). According to the variables shown in Table II, in the form "name_D-i" for data corresponding to i days before today, the input attribute set contains 553 items.

$$21 \text{ initial vars.} + (19 \text{ historical vars./day} \times 28 \text{ days}) = 553 \text{ variables}$$

Calendar data are not considered between historical variables, this is because the number of initial variables differs.

3.2 Initial Features Selection and Fuzzification

In order to reduce the initial variable input space, PCA (Principal Components Analysis) [10] [11] has been used. The initial 553 characteristics were reduced to 295, considering 90% of explained variance and the values over the average of the eigenvectors.

The optimal fuzzy partition for each feature was calculated using KBCT [12]. This tool generates 3 different types of partitions for each variable: *HFP* [13] [14], *Regular* and *Kmeans* [15], with a [2, 9] range for the number of different linguistic labels. The partition that got the best ratings was selected, according to the following indexes and criteria: minimizes *Partition Entropy* (PE) [16] and maximizes *Partition Coefficient* (PC) [16] and *Chen index* [17].

Table 2. KBCT results summary for input fuzzification

| Variable | Labels | Chen index | Method | Partition Centers |
|----------------|--------|------------|---------|-----------------------|
| No.Inc | 3 | 0.7556 | Kmeans | (77, 82.5, 87) |
| Information | 3 | 0.9865 | HFP | (3.5, 11.5, 16.3) |
| Incidents | 3 | 0.7836 | Kmeans | (11.5, 17, 22.5) |
| Sanitary | 3 | 0.7314 | Kmeans | (49, 56, 64) |
| Security | 3 | 0.7165 | Kmeans | (28, 35.5, 42.5) |
| SAR | 3 | 0.9473 | HFP | (7, 22.5, 24.5) |
| Basic_Serv. | 3 | 0.9189 | HFP | (2.1, 7.2, 9.9) |
| Call_Midnight | 3 | 0.7657 | Kmeans | (98, 205, 345) |
| Call_Dawn | 3 | 0.9625 | HFP | (50, 165, 214) |
| Call_Morning | 3 | 0.8736 | HFP | (240, 610, 722) |
| Call_Afternoon | 3 | 0.7798 | Kmeans | (230, 400, 560) |
| Call_Evening | 3 | 0.8717 | HFP | (230, 650, 770) |
| Total_Calls | 3 | 0.7967 | Kmeans | (3100, 5200, 7050) |
| T_Avg | 3 | 0.7849 | Kmeans | (4, 12, 22) |
| T_Max | 3 | 0.7758 | Kmeans | (10, 18.5, 30.5) |
| T_Min | 3 | 0.7663 | Kmeans | (-4, 3, 11) |
| Rain | 3 | 0.9816 | HFP | (30, 110, 192.2) |
| Snow | 7 | 1.0000 | Regular | (1, 2, 3, 4, 5, 6, 7) |
| Fog | 7 | 1.0000 | Regular | (1, 2, 3, 4, 5, 6, 7) |

3.3 FUDT Construction

The limits of inhomogeneity and segment size threshold are established experimentally for each problem. The lower limit is set at 4 elements, for a lower number segment could be considered irrelevant and subject to the effects of any possible outlier or noise. On the other hand, due to the total number of samples, over 30 items by segment means that the tree would collapse, do not creating any node other than root.

3.4 Experimental Results of FUDT

Table 4 shows the main results obtained with the FUDT algorithm using the selected fuzzy variables described in Subsection 3.2 above and the thresholds of Table 3. Columns correspond with the number of Leaf Nodes or terminals (LN) and Depth (D) of the tree in each experiment, while IT is for Inhomogeneity Threshold and SST for Segment Size Threshold. They are grouped into two blocks, showing quality measures depending on the structure of the trees: T , $IQN-T$ and CI , and the quality indexes of the clusters: SC , S and XB .

In order to compare values from SC , S and XB , these values have been divided by the number of leaf nodes, due to the fact that each tree has a different number of nodes.

It has been found that beyond a certain inhomogeneity threshold level, the tree collapses, thus not creating any node other than the root node. For this reason, these values have not been considered and the study focuses on the really relevant cases. Therefore, the tree has been rebuilt by fixing a value for the inhomogeneity threshold and giving values to the segment size threshold. This avoids the influence of extreme values on the index normalization. Table 4 contains the renewed indexes and Figure 1 shows the table data graphically.

Recall that the measure T gives a higher value on those trees that are more compact, reaching the maximum ($T \simeq 1$) when all the leaf nodes derivate directly

Table 3. Thresholds for the construction of the fuzzy UDT

| Parameter | Lower Limit | Upper Limit | Step Num. | Values |
|------------------------------|-------------|-------------|-----------|--------|
| Inhomogeneity Threshold (IT) | 0.16 | 0.19 | 0.01 | 4 |
| Segment Size Threshold (SST) | 4 | 30 | 2 | 14 |

Table 4. Results of Fuzzy UDT based on quality indexes

| IT | SST | LN | D | T | IQN-T | IC | SC | S | XB |
|------|-----|----|---|--------|--------|--------|--------|--------|--------|
| 0.40 | 4 | 48 | 9 | 0.0000 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.40 | 6 | 40 | 7 | 0.1454 | 0.9193 | 0.2262 | 0.0726 | 0.0737 | 0.1659 |
| 0.40 | 8 | 33 | 5 | 0.3041 | 0.8139 | 0.4902 | 0.0895 | 0.0985 | 0.1347 |
| 0.40 | 10 | 31 | 5 | 0.2777 | 0.7664 | 0.5112 | 0.1136 | 0.1233 | 0.1858 |
| 0.40 | 12 | 26 | 5 | 0.4008 | 0.6459 | 0.7550 | 0.1897 | 0.1886 | 0.3414 |
| 0.40 | 14 | 24 | 5 | 0.4516 | 0.5882 | 0.8634 | 0.2012 | 0.2096 | 0.4155 |
| 0.40 | 16 | 22 | 5 | 0.5324 | 0.5054 | 0.9731 | 0.2808 | 0.2792 | 0.4530 |
| 0.40 | 18 | 20 | 5 | 0.5957 | 0.4230 | 0.8273 | 0.2825 | 0.2862 | 0.6176 |
| 0.40 | 20 | 18 | 5 | 0.6674 | 0.3266 | 0.6592 | 0.3370 | 0.3512 | 0.6993 |
| 0.40 | 22 | 17 | 5 | 0.6674 | 0.2615 | 0.5941 | 0.4822 | 0.5076 | 0.7622 |
| 0.40 | 24 | 17 | 5 | 0.6674 | 0.2615 | 0.5941 | 0.4822 | 0.5076 | 0.7622 |
| 0.40 | 26 | 15 | 5 | 0.8643 | 0.1616 | 0.2973 | 0.8002 | 0.8162 | 0.8819 |
| 0.40 | 28 | 14 | 5 | 0.9241 | 0.0729 | 0.1488 | 0.9331 | 0.9685 | 0.8938 |
| 0.40 | 30 | 13 | 5 | 1.0000 | 0.0000 | 0.0000 | 1.0000 | 1.0000 | 1.0000 |

from the root. On the other hand, the value of $IQN-T$ is maximum when the tree is fully expanded, when it reaches the minimum value of impurity. Since a too expanded tree can lose efficiency due to the effect of overfitting and too simple ones can have a very high classification error, the IC was suggested as a way to find a trade-off between both values.

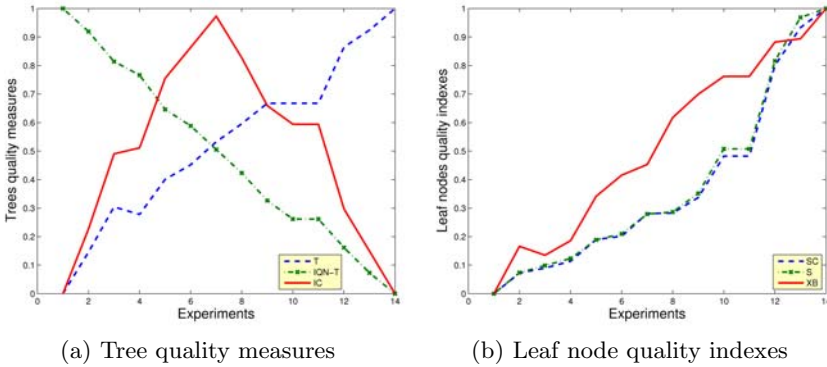


Fig. 1. Tree and cluster quality indexes

Since a low value for SC indicates a better partition, and the optimal number of clusters should minimize the value of the Xie and Beni index (XB), from Figure 1(b), it is possible to deduce that the best trees are the most widespread since the first experiments have a smaller segment size threshold. As previously, these trees can suffer from weak generalization capability.

The maximum value for the IC (Figure 1(a)) is achieved by an inhomogeneity threshold of 0.40 and a segment size of 16. For this experiment, the value of SC and XB is reasonably low, so this seems a good candidate for further study. The schema below shows the internal structure of the selected tree.

The tree has 37 nodes, 22 of them are leaf or terminal nodes and the tree depth is 5. This means that at most there are 22 different classes that are described by rules generated from the root node to each of the leaves, which at most have 5 antecedents. These rules are compatible with readability-interpretability criteria, because 7 ± 2 is considered in scientific literature as a human limit in order to achieve an efficient handling of the rules [18].

The most populated nodes are 26 and 30. Their descriptions, according to linguistic attributes, point out that the first one corresponds with soft temperature days, a low number of total calls distributed in a high number of calls related to *Incidents* but a low number of calls related to *No Incidents*. The second node however describes days characterized by high temperatures, a low number of calls, but equally distributed between *Incidents* and *No Incidents*.

Fuzzy UDT (IT: 0.40, SST: 16).

```

-->(1) Calls_D-14
  +-[low]->(2) T_avg_D-11
    |
    | +-[low]-->(4) Calls_D-26
    | |
    | | +-[low]----->(10) Incidents_D-16
    | | |
    | | | +-[low]-->(23) Month
    | | | |
    | | | | +-[Jan,Feb]->(32) Fog_D-28
    | | | | |
    | | | | +-[Mar]----->(33) Call_Morning_D-03
    | | | | |
    | | | | +-[Nov,Dic]->(34) Call_Morning_D-20
    | | | | |
    | | | | +-[med]-->(24) Month
    | | | | |
    | | | | +-[high]->(25) Month
    | | | | |
    | | | | | +-[Jan]----->(35) Fog_D-24
    | | | | | |
    | | | | | +-[Feb,Mar]->(36) Fog_D-28
    | | | | | |
    | | | | | +-[Nov,Dic]->(37) T_min_D-03
    | | | |
    | | | +-[med,high]->(11) Month
    | | +-[med]-->(5) Calls_D-04
    | | |
    | | | +-[low]----->(12) Calls_D-02
    | | | |
    | | | | +-[low]----->(26) Calls_D-26
    | | | | |
    | | | | +-[med,high]->(27) Month
    | | | | |
    | | | | +-[med,high]->(13) T_avg_D-27
    | | | | |
    | | | | | +-[low]----->(28) Call_Morning_D-26
    | | | | | |
    | | | | | +-[med,high]->(29) Month
    | | | |
    | | +-[high]->(6) Calls_D-10
    | | |
    | | | +-[low]----->(14) Calls_D-17
    | | | |
    | | | | +-[low]----->(30) Calls_D-05
    | | | | |
    | | | | +-[med,high]->(31) Call_Evening_D-05
    | | | | |
    | | | | +-[med,high]->(15) Call_Midnight
    | | +-[med,high]->(3) T_avg_D-22
    | | |
    | | | +-[low]-->(7) Month
    | | | |
    | | | | +-[Jan,Feb]----->(16) Fog_D-24
    | | | | |
    | | | | +-[Mar,Apr,May,Nov,Dic]->(17) Call_Morning_D-07
    | | | | |
    | | | | +-[med]-->(8) Month
    | | | | |
    | | | | | +-[Jan,Feb,Mar,Apr]->(18) Call_Midnight_D-09
    | | | | | |
    | | | | | +-[May,Jun]----->(19) Call_Evening_D-01
    | | | | | |
    | | | | | +-[Sep,Oct,Nov,Dic]->(20) Call_Midnight_D-25
    | | | | | |
    | | | | | +-[high]->(9) Call_Midnight_D-03
    | | | | | |
    | | | | | | +-[low]----->(21) Call_Midnight_D-24
    | | | | | | |
    | | | | | | +-[med,high]->(22) Call_Midnight
  
```

4 Conclusions

A fuzzy version of the unsupervised decision tree algorithm has been considered in order to deal with the assignments of resources for a call center. The proposal is based on the characterization of day categories in order to determine the workload and resources needed for each category. Due to the weak expert knowledge available on this way of facing the problem, an unsupervised approach have been considered. These categories have to be described in linguistic terms on the domain concerned.

A selection of variables has been carried out in two steps: using PCA to obtain a pre-selection and Fuzzy UDT, which decides the final variables to be taken into account. This two step methodology is mandatory due to the high number of initial variables considered and the unsupervised tree approach.

In order to obtain a reasonable tree, a hybrid solution based on indexes of tree and cluster quality have been used. The results obtained can be considered reasonable, and in future versions, they can be improved by the refinement of the criteria considered. An optimized version of the knowledge base could be developed, but the level of interpretability must be kept.

The developed system intends to help the managers of the emergency call center, establishing the base line for the normal operation of the system. Later, with some estimations and forecasts of certain variables, the system will characterize the desired day, which will help in the decision making of planning the resources assignment.

References

1. Mandelbaum, A., Garnett, O., Reiman, M.: Designing a call center with impatient customers. *Manufacturing and Service Operations Management* 4(3), 208–227 (2002)
2. Pajares, R.G., Benitez, J.M., Palmero, G.S.: Feature Selection for Time Series Forecasting: A Case Study. In: 8th International Conference on Hybrid Intelligent Systems, pp. 555–560 (2008)
3. Basak, J., Krishnapuram, R.: Interpretable hierarchical clustering by constructing an unsupervised decision tree. *IEEE Transactions on Knowledge and Data Engineering* 17(1), 121–132 (2005)
4. Wu, W., Kumar, V.: *The Top Ten Algorithms in Data Mining*. CRC Press, Boca Raton (2009)
5. Mitra, S., Acharya, T.: *Data mining: multimedia, soft computing, and bioinformatics*. John Wiley, Chichester (2003)
6. Fournier, D., Cremilleux, B.: A quality index for decision tree pruning. *Knowledge-Based Systems* 15, 37–43 (2002)
7. Quinlan, J.R.: Simplifying decision trees. *Int. J. Man-Mach. Stud.* 27(3), 221–234 (1987)
8. Bensaid, A.M., Hall, L.O., Bezdek, J.C., Clarke, L.P., Silbiger, M.L., Arrington, J.A., Murtagh, R.F.: Validity-guided (Re)Clustering with applications to image segmentation. *IEEE Transactions on Fuzzy Systems* 4, 112–123 (1996)
9. Xie, X.L., Beni, G.A.: Validity measure for fuzzy clustering. *IEEE Trans. PAMI* 3(8), 841–846 (1991)
10. Mao, K.Z.: Identifying critical variables of principal componentes for unsupervised feature selection. *IEEE T. on Systems, Man, and Cybernetics* 35(2), 339–344 (2005)
11. Malhi, A., Gao, R.X.: Pca-based feature selection scheme for machine defect classification. *IEEE T. on Instrumentation and Measurement* 53(6), 1517–1525 (2004)
12. KBCT: Knowledge Base Configuration Tool, <http://www.mat.upm.es/projects/advocate/en/index.html>
13. Guillaume, S., Charnomordic, B.: A new method for inducing a set of interpretable fuzzy partitions and fuzzy inference systems from data. *Studies in Fuzziness and Soft Computing*, pp. 148–175. Springer, Heidelberg (2003)
14. Guillaume, S., Charnomordic, B.: Generating an interpretable family of fuzzy partitions. *IEEE Transactions on Fuzzy Systems* 12(3), 324–335 (2004)
15. Hartigan, J.A., Wong, M.: A k-means clustering algorithm. *Applied Statistics* 28, 100–108 (1979)
16. Bezdek, J.C.: *Pattern recognition with fuzzy objective functions algorithms*. Plenum Press, New York (1981)
17. Chen, M.Y.: Establishing interpretable fuzzy models from numeric data. In: *Proceedings of the 4th World Congress on Intelligent Control and Automation*, pp. 1857–1861 (2002)
18. Zhou, S., Ganb, J.Q.: Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. *Fuzzy Sets and Systems* 159, 3091–3131 (2008)

Optimization of Embedded Fuzzy Rule-Based Systems in Wireless Sensor Network Nodes

Manuel-Ángel Gadeo-Martos¹, Jose-Ángel Fernández-Prieto¹, Joaquín Canada Bago¹,
and Juan-Ramón Velasco²

¹Telecommunication Engineering Department
Universidad de Jaén, Spain

{gadeo, jan, jcbago}@ujaen.es

²Department of Automatic
Universidad de Alcalá, Spain
juanra@aut.uah.es

Abstract. Nowadays, growing interest exists on the integration of artificial intelligence technologies, such as neural networks and fuzzy logic, into Wireless Sensor Networks. However, few attentions have been paid to integrate knowledge based systems into such networks. The objective of this work is to optimize the design of a distributed Fuzzy Rule-Based System embedded in Wireless Sensor Networks. The proposed system is composed of: a central computer, which includes a module to carry out knowledge bases edition, redundant rules reduction and transformation of knowledge bases with linguistic labels in others without labels; access point; sensor network; communication protocol; and Fuzzy Rule-Based Systems adapted to be executed in a sensor. Results have shown that, starting from knowledge bases generated by a human expert, it is possible to obtain an optimized one with a design of rules adapted to the problem, and a reduction in number of rules without a substantial decrease in accuracy. Results have shown that the use of optimized knowledge bases increases the sensor performance, decreasing their run time and battery consumption. To illustrate these results, the proposed methodology has been applied to model the behavior of agriculture plagues.

Keywords: Fuzzy Rule-Based Systems, Wireless Sensor Networks.

1 Introduction

Wireless Sensor Networks (WSNs) [1] have become an enabling technology for a wide range of applications. The traditional WSN architecture consists of a large number of sensor nodes which are densely deployed over an area of interest. These nodes can be conceived as small computers, extremely basic in terms of their interfaces and their components [2], having limited battery, reduced memory and processing capabilities.

On the other hand, there are some tendencies to include artificial intelligent technologies in WSNs [3][4][5], such as artificial neural network and fuzzy logic. However, few attentions have been paid to integrate Fuzzy Rule-Based Systems (FRBSs) into WSNs (embedded FRBSs). In [7] and [6, 8], two schemes have been proposed

for embedded FRBSs, applied to model fire detection and an agriculture plague respectively. These schemes do not describe the use of a human-machine interface neither a FRBS specifically adapted to sensor limitations. In order to make easy and efficient the integration of embedded FRBSs in WSNs, the structure proposed in this work includes an interface to make possible the knowledge base (KB) edition and obtaining operation results, and a FRBS designed to increase sensor performance. On the other hand, the use of an interface makes easy this work but it does not guarantee an optima KB generation. In addition, this paper introduces two additional modules proposed to optimize these KBs by means of rule redundancy reductions and rule adaptations to specific modeling problems.

The remainder of the paper is organized as follows. Section 2 presents a brief description of the distributed architecture proposed. Section 3 describes the functionality of each module included in this architecture. Results are reported in Section 4 and finally some conclusions are drawn in Section 5.

2 Distributed Architecture for Fuzzy Rule-Based System Embedded in Wireless Sensor Network

Basically, the factors that determinate the embedded FRBS performance are three: 1) the limitations associated with physical resources; 2) the ones related with an appropriated FBRS structuring and programming; 3) the associated with a KB right design to be used. An appropriate FBRS performance will decrease the consumption of energy system, what will increase the time of battery discharge. To optimize this performance, this work proposes the use of a FBRS with distributed functions placed in: 1) central computer, the tasks of KB edition and optimization; and 2) nodes into WSN, the modules of input scaling, fuzzyfication, inference engine which use the optimized KB, defuzzyfication, output scaling, and communication module (figure 2).

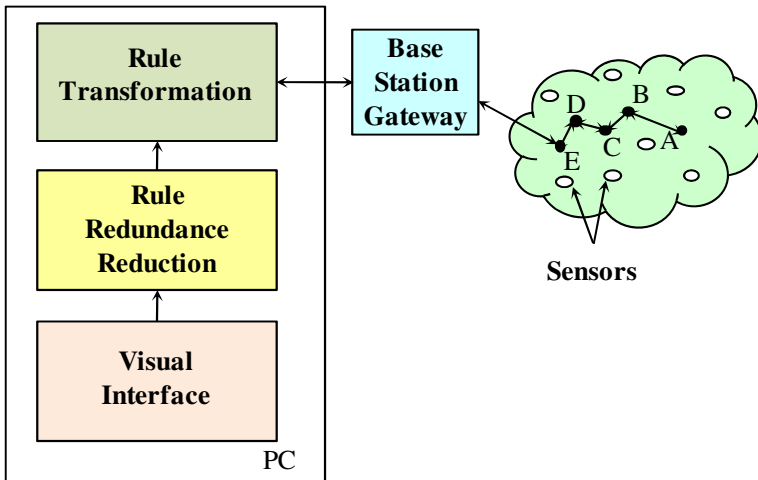


Fig. 1. General structure of the system

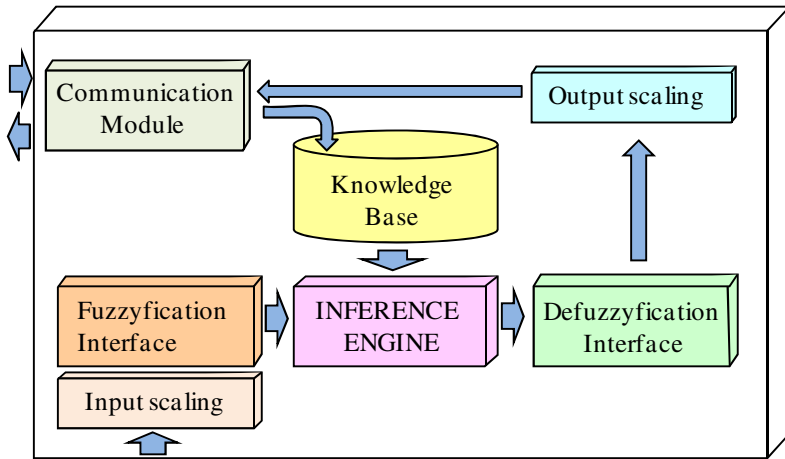


Fig. 2. Basic structure of a Mamdani-FBRS embedded into a sensor

The process of KB optimization proposed in this paper does not deal with the generation of an optima KB, rather it is concerned with the transformation of a given KB, in order to decrease the inference algorithm run time and the energy consumption, with a reasonable decrease in their accuracy. The KB optimization goal is to obtain a KB that allows the increase in the sensor global performance.

The proposed design, specially the inference engine, has been adapted to physical limitations of the used sensor: Sun SPOT [9] (180 MHz 32-bit ARM920T processor, 512K RAM and 802.15.4 radio).

Figure 1 shows the general structure of the system, which is composed of: personal computer, base station gateway, communication protocol, and sensors in WSN.

3 Fuzzy Rule-Based System Embedded in Sensor Modular Structure Description

The FRBS used in this paper is based on the model of Mamdani [10]. Two variants of Mamdani FRBSs have been proposed [11]: 1) descriptive and 2) approximate [12] [13] [14] [15]. The structures of these systems are similar, but each type of this FRBS has different properties and presents complementary advantages and drawbacks. In order to improve embedded FRBS behavior, it is possible to use these advantages.

In descriptive FBRS (or linguistic Mamdani FRBS), rules carry a linguistic label that points to a particular fuzzy set of a linguistic partition of the underlying linguistic variable. In approximate FBRS rules, the input variables and the output one are fuzzy variables instead of linguistic variables.

Approximate FRBSs demonstrate some specific advantages over linguistic FRBSs making them particularly useful for certain types of applications [14]: 1) each rule employs its own distinct fuzzy sets, resulting in additional degrees of freedom and increase in expressiveness and 2) the number of rules can be adapted to the complexity of the problem. These properties enable approximate FRBSs to achieve a better

degree of accuracy than linguistic FRBS in complex problem domains. In descriptive FRBSs, the main advantage is the large degree of interpretability of linguistic rules [16] [17] [18] [19].

In this paper, we propose the use of a visual interface as a method of KB composition. Using this interface, the human expert can specify the linguistic labels associated to each linguistic variable, the structure of the rules in the rule base (RB), and the meaning of each label. This method is the simplest one to be applied when the expert is able to express his knowledge in the form of linguistic rules. To improve the accuracy of this linguistic approach, this interface enables the specification of exact membership functions as well.

In the process of KB optimization it is necessary to avoid redundant rules. These If-Then rules have the property that a system state is covered by more than one rule as the fuzzy sets in the antecedents overlap. The existence of redundant rules may cause degradation in the performance of the FBRS. Therefore, it is important to evaluate the utility of a rule, by analyzing its impact on the global system behavior, and then decide whether a rule should be discarded from the rule set.

In the approach proposed in this paper, after the KB is composed, a module is used to reduce rules redundancy. Firstly, its algorithm searches rules with overlap in their antecedent and consequent; secondly, it makes groups of rules with only one difference in the same proposition of the antecedent; and thirdly, for each one of these groups, it selects the rules with adjacent fuzzy sets in this proposition. These redundant rules can be simplified in only one rule. The new rule will be the same as the old rules except in their different proposition, and now, their new fuzzy set will be built by the composition of the involved adjacent fuzzy sets. A new KB is made with reduction of redundant rules.

A reduction of rules can produce lack of accuracy in modeling. In order to evaluate the accuracy of a new KB, this paper proposes the use of an algorithm that contains the following steps: first, for a wide set of system states, it obtains the output of FRBS, using two KB: 1) the original (output 1) and 2) the reduced one (output 2); second, for each FRBS output it determines the absolute value of the subtraction (output 1 – output 2); and third it sums these absolute errors. In the interface, the last module enables the analysis of this error, to decide if the reduced KB achieves the target of accuracy.

In order to make the most of approximate and descriptive FRBS advantages, this approach enables the use of a visual interface to make easy the composition of the descriptive and approximate KBs, while the inference engine, of the embedded FRBS, works as the approximate Mamdani-type. Therefore, previously at the KB transmission, it is necessary the transformation of the descriptive KB into an approximate one.

In order to reduce the computational burden this paper proposes the use of FRBS in mode B-FITA (First Infer, Then Aggregate) and the operator centre of gravity. In the Sun SPOT sensor, the embedded inference engine has been programmed in Java using the J2ME platform.

4 Results

In order to illustrate the proposed methodology, three experiments (Ex) have been performed to model, with simple KBs, the behavior of two plagues of the olive tree:

the Prays (Prays oleae bern) and Repilo (Spilocaea oleagina). The life cycles of these plagues depend on humidity and temperature, but they are not the same. The common steps of each experiment consist of: KB generation in visual interface, rule transformations, KB transmission, modeling a wide set of system states (4000) into FRBS embedded, and transmission of results. Furthermore, each experiment has a specific part. In Ex 1, it has been used an approximate KB which is composed of a RB with two groups of rules (GR) (table 1 and 2) and theirs associated specific fuzzy set (FS) definitions (Fig 3 and 4).

In Ex 2, it has been used an approximate KB which is composed of a RB with two GR (table 3 and 4) and theirs associated specific FS definitions (Fig 6 and 7). In this case, the rules and FS definitions have been obtained using the redundancy reduction module. As can be observed this algorithm decreases six rules (table 3 and 4) and generates, by means of fusion, three new FSs (figure 6 and 7).

With the purpose of illustrating the redundancy reduction process, an example of redundant rule reduction is shown. In the group of rules used to model the Prays (table 1), it can be noted a set of three rules with the same consequent and the same second proposition in their antecedent:

R1: If Humidity is S and Temperature is VS Then Prays Alert is VS

R6: If Humidity is M and Temperature is VS Then Prays Alert is VS

R11: If Humidity is L and Temperature is VS Then Prays Alert is VS

Table 1. Group of rules used to model the Prays (in specific and common KB)

| <i>Group of rules used in the model of the "Prays"</i> | | | | | | |
|--------------------------------------------------------|--------------------|----|----|----|---|----|
| <i>Prays Alert</i> | <i>Temperature</i> | | | | | |
| | | VS | S | M | L | VL |
| <i>Humidity</i> | S | VS | VS | S | S | VS |
| | M | VS | S | M | M | S |
| | L | VS | M | VL | L | M |

Table 2. Group of rules used to model the Repilo (in specific and common KB)

| <i>Group of rules used in the model of the "Repilo"</i> | | | | | | |
|---------------------------------------------------------|--------------------|----|---|----|---|----|
| <i>Repilo Alert</i> | <i>Temperature</i> | | | | | |
| | | VS | S | M | L | VL |
| <i>Humidity</i> | S | VS | S | S | S | VS |
| | M | S | M | L | M | S |
| | L | M | L | VL | L | M |

Table 3. Reduced group of rules used to model the Prays

| <i>Reduced Group of rules used in the model of the “Prays”</i> | | | | | | |
|----------------------------------------------------------------|--------------------|----|----|----|---|----|
| <i>Prays Alert</i> | <i>Temperature</i> | | | | | |
| <i>Humidity</i> | | VS | S | M | L | VL |
| | S | VS | VS | S | | VS |
| | M | | S | M | | S |
| | L | | M | VL | L | M |

Table 4. Reduced group of rules used to model the Repilo

| <i>Reduce Group of rules used in the model of the “Repilo”</i> | | | | | | |
|----------------------------------------------------------------|--------------------|----|---|----|---|----|
| <i>Repilo Alert</i> | <i>Temperature</i> | | | | | |
| <i>Humidity</i> | | VS | S | M | L | VL |
| | S | VS | S | | | VS |
| | M | S | M | L | M | S |
| | L | M | L | VL | L | M |

Applying the proposed algorithm these original rules can be simplified in only one rule (New R1).

New R1: If Humidity is S&M&L and Temperature is VS Then Prays Alert is VS

Except for the first proposition in the antecedent, the rule “New R1” is equal to the three originals. Now, the FS associated with the “Humidity” variable is “S&M&L” (figure 6), which has been obtained by means of composition of the three original FS (figure 3).

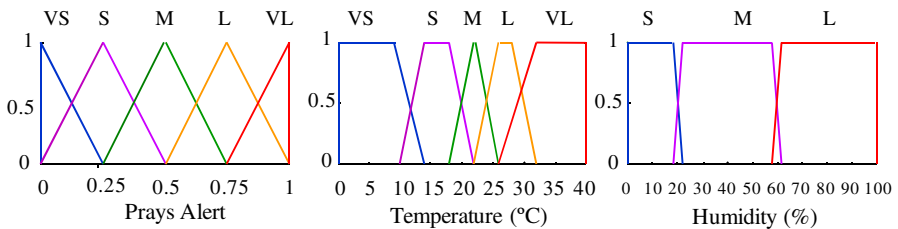


Fig. 3. Membership functions of the inputs and output variables fuzzy sets defined in specific approximate rules for Prays modeling

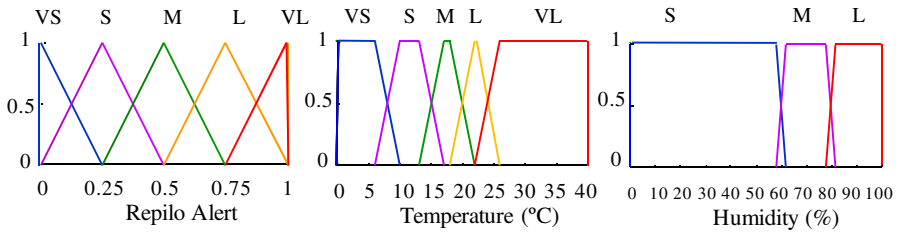


Fig. 4. Membership functions of the inputs and output variables fuzzy sets defined in specific approximate rules for Repilo modeling

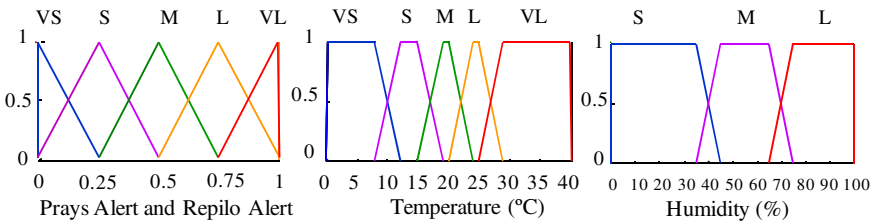


Fig. 5. Membership functions of the inputs and output variables fuzzy sets included in common descriptive KB for Prays and Repilo modeling

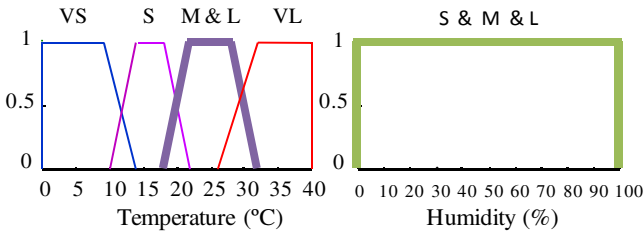


Fig. 6. Input fuzzy sets for Prays modeling, in reduced KB

In Ex 3, it has been used a descriptive KB which is composed of a RB with two GRs (table 1 and 2) and the definition of linguistic labels associated to each linguistic variable (Fig 5). In this case, the use of linguistic labels prevents specific FS definitions for each GR. Once the three experiments have been made, the following parameters have been calculated: accuracy of plague modeling, run time (inference rate) and battery consumption (charge of battery) performances (table 5).

In table 5, the accuracy parameter has been obtained by means of division of absolute error addition (in Prays and Repilo output) by the sum of descriptive FRBS Prays and Repilo output. As can be observed in table 5: a) the use of specific approximate FRBS to model the Prays and Repilo olive plagues provides an improvement in accuracy,

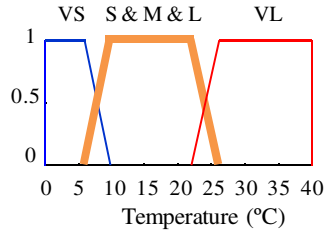


Fig. 7. Input fuzzy sets for Repilo modeling, in reduced KB

Table 5. Comparison of performance considering different types of embedded FRBS

| FRBS | Execution of 4000 continued inferences | | | | |
|---------------------|----------------------------------------|--------------|------------------|--------------|--------------|
| | Time (ms) | Increase (%) | Consumption (mA) | Increase (%) | Accuracy (%) |
| Approximate | 17716,6 | 0,00 | 0,5666 | 0,00 | 0,00 |
| Descriptive | 17177,7 | -3,04 | 0,5469 | -3,48 | -28,15 |
| Approximate Reduced | 14080,3 | -20,52 | 0,4501 | -20,56 | -0,06 |

compared with the use of descriptive FRBS; b) the use of specific approximate FRBS, with reduced KB, provides an improvement in the run time (or inference rate) and the battery consumption (or time of battery discharge), compared with the use of specific approximate FRBS. From the analysis of the experimental results obtained, we notice that, it has been possible to compose an optimized KB, adapted to model olive plagues, which generates a decrease in FRBS run time (increase the inference rate) and a decrease in battery consumption (decrease the time of discharge in battery sensor), without decreasing the accuracy.

5 Conclusions

In this paper, we have presented a distributed structure of FRBS embedded in WSN which incorporates: a) a visual interface to make easy the composition of approximate and descriptive KBs; b) a module to reduce rule redundancy; c) a module to transform descriptive into approximate KBs; d) a communication protocol; and e) an approximate FRBS adapted to be executed in a sensor. Results have shown the effectiveness of the proposed structure to optimize the execution of FRBS embedded into a sensor.

Acknowledgments. This work has been partially supported by Ministerio de Ciencia e Innovacion (project TEC2009-13619), Spain.

References

1. Karl, H., Willig, A.: *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, Chichester (2005)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cyirci, E.: Wireless sensor networks: A survey. *Computer Networks* 38(4), 393–422 (2002)
3. Karlsson, B.: *Intelligent Sensor Networks - an Agent-Oriented Approach*. In: *Workshop on Real-World Wireless Sensor Networks* (2005)
4. Kulakov, A., Davcev, D.: Tracking of unusual events in wireless sensor networks based on artificial neural-networks algorithms. In: *Proceedings of the International Conference on Information Technology: Coding and Computing*. IEEE, Los Alamitos (2005)
5. Averkin, A.: Soft Computing in WSNs. In: *Proceedings of the EUSFLAT*, pp. 387–390 (2007)
6. Cañada-Bago, J.: From a genetic fuzzy rule-based system to an intelligent sensor network. In: *Proceedings of International Conference on Sensor Technologies and Applications*, pp. 373–377. IEEE, Valencia (2007)
7. Marin-Perianu, M., Havinga, P.: D-FLER: A distributed fuzzy logic engine for rule-based wireless sensor networks. In: *International Symposium on Ubiquitous Computing Systems (UCS)*, pp. 86–101 (2007)
8. Cañada-Bago, J., Gadeo-Martos, M.A., Fernández-Prieto, J.A., Velasco, J.R.: Poster Abstract: A Knowledge Based Wireless Sensor Network. In: *Proceeding of European Wireless Sensors Network (EWSN 2009) – Demos/Posters Session*, Cork, Ireland, pp. 21–22 (2009)
9. Sun Microsystems. Home of Project Sun SPOT, <http://www.sunspotworld.com/>
10. Mamdani, E.H.: Applications of fuzzy algorithm for control a simple dynamic plant. *Proceedings of the IEE* 121(12), 1585–1588 (1974)
11. Cordon, O., Herrera, F., Hoffmann, F., Magdalena, L.: *Genetic Fuzzy Systems: Evolutionary tuning and learning of fuzzy knowledge bases*. *Advances in fuzzy systems – Applications and theory*, vol. 19. World scientific Publishing, Singapore (2001)
12. Bardossy, A., Duckstein, L.: *Fuzzy Rule-Based Modeling with Application to Geographical, Biological and Engineering Systems*. CRC Press, Boca Raton (1995)
13. Cordon, O., Herrera, F.: A general study on genetic fuzzy systems. In: Periaux, J., Winter, G., Galán, M., Cuesta, P. (eds.) *Genetic Algorithms in Engineering and Computer Science*, pp. 33–57. John Wiley and Sons, Chichester (1995)
14. Carse, B., Fogarty, T.C., Munro, A.: Evolving fuzzy rule based controllers using genetic algorithms. *Fuzzy Sets and Systems* 80, 273–294 (1996)
15. Koczy, L.: Fuzzy if ... then rule models and their transformation into one other. *IEEE Transactions on Systems, Man, and Cybernetics* 26(5), 621–637 (1996)
16. Driankov, D., Hellendoorn, H., Reinfrank, M.: *An introduction to Fuzzy Control*. Springer, Heidelberg (1993)
17. Lee, C.C.: Fuzzy logic in control systems: fuzzy logic controller- Parts I and II. *IEEE Transactions on Systems, Man, and Cybernetics* 20(2), 404–418, 419–435 (1990)
18. Pedrycz, W.: *Fuzzy Modelling: Paradigms and Practice*. Kluwer Academic Publishers, Dordrecht (1996)
19. Sugeno, M., Yasura, T.: A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems* 1(1), 7–31 (1993)

An Algorithm for Online Self-organization of Fuzzy Controllers

Ana Belén Cara, Héctor Pomares, and Ignacio Rojas

Dept. of Computer Technology and Computer Architecture, University of Granada
Periodista Saucedo Aranda s/n, 18071 Granada, Spain
{acara,hector,irojas}@atc.ugr.es

Abstract. This work presents a fuzzy controller capable of designing its own structure online, based on the data obtained during the normal system operation. The algorithm does not use previous information about the differential equations that define the plant's behaviour. The controller may be initially empty, as the method is able to distinguish the important input variables by assigning them more membership functions. With this aim, the method works in two stages: the adaptation of the consequents for every tested topology and the online addition of new membership functions. To show its capabilities, simulation results with a liquid tank system are analyzed.

Keywords: adaptive control, intelligent systems.

1 Introduction

The popularity of fuzzy controllers over the last years is highly due to its main capabilities: They avoid the need of accurate mathematical models of the systems under control and they make it possible to apply human expert knowledge about the operation of such systems in order to design a proper controller [1]; besides, it has been proved that they are universal approximators (i.e. they are able to approximate any continuous function in a compact set to any desired accuracy) [2]. The design of a fuzzy controller is not always an easy task [3]. Ad-hoc fuzzy controllers can be designed for experts when the plant under control is well known [4]. On the other hand, intelligent and automatic methods are needed when the knowledge about the plant is more reduced. The literature shows several methods to adapt the controller's parameters online (i.e. consequents of the rules and/or membership functions) [5] or even their topology [3,1]. However, the aforementioned methods rely on assumptions made about the plant's equations (e.g. certain bounds are supposed to be known), so their application is not always possible. Therefore, the most challenging case is being unable to make such assumptions. This problem has been solved offline, with algorithms based on pretraining with I/O data [6]. The online adaptation of the controller's parameters for fixed topologies has been addressed as well [7]. However, little has been written about the online self-organization of the fuzzy controller when no prior knowledge about the plant is available [8].

In this work we present a method for the online self-organization of the topology of a fuzzy controller, together with the adaptation of the rule consequents. Both types of adaptation are performed while the controller is working, providing better tolerance to noise and robustness under changes in the plant's dynamics. The algorithm is based in the property of universal approximation of fuzzy controllers, which states that the proper addition of membership functions makes possible to reach a desired accuracy level for a functional approximation. To illustrate its capabilities, simulation results are provided.

2 Online Self-organization of Fuzzy Controllers

The mathematical representation of the plant to be controlled may be expressed in terms of its differential equations or by its difference equations, provided that these are obtained from the former using a short enough sampling period:

$$y(k + 1) = f(y(k), \dots, y(k - p), u(k), \dots, u(k - q)) \tag{1}$$

where $y(k)$ is the plant output at time k , f is an unknown continuous and differentiable function, u is the control input and p and q are constants which determine the order of the plant. The restriction usually imposed to these plants is that they must be controllable [7], i.e. $\frac{\partial f}{\partial u} \neq 0$ for every input within the operation range. Hence, the mentioned derivative must have a constant sign [3,7]. The aim of the controller is to guarantee that the plant's output tracks a given reference signal, $r(k)$. Thus, in the absence of actuator bounds, we can assume that there exists a function F such that the control input given by

$$u(k) = F(\mathbf{x}(k)) \tag{2}$$

with $\mathbf{x}(k) = (r(k), y(k), \dots, y(k - p), u(k - 1), \dots, u(k - q))$, is capable of reaching the set point in the following instant: $y(k + 1) = r(k)$. In order to approximate such function we employ a 0-order TSK fuzzy system with a complete set of rules that are defined as

$$\text{IF } x_1 \text{ is } X_1^{i_1} \text{ AND } x_2 \text{ is } X_2^{i_2} \text{ AND } \dots x_N \text{ is } X_N^{i_N} \text{ THEN } u = R_{i_1 i_2 \dots i_N} \tag{3}$$

where $X_v^{i_v} \in \{X_v^1, X_v^2, \dots, X_v^{n_v}\}$ are the membership functions of input X_v , n_v is the number of membership functions for that variable and $R_{i_1 i_2 \dots i_N}$ is a numerical value representing the rule consequent. The characteristics of the fuzzy system are: triangular membership functions (MF), product as T-norm for the inference method and weighted average for defuzzification. Thus, the fuzzy controller's output is given by:

$$u(k) = \hat{F}(\mathbf{x}(k); \Theta) = \frac{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_N=1}^{n_N} R_{i_1 i_2 \dots i_N} \cdot \prod_{m=1}^N \mu_{X_m^{i_m}}(x_m(k))}{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_N=1}^{n_N} \prod_{m=1}^N \mu_{X_m^{i_m}}(x_m(k))} \tag{4}$$

where $\mu_{X_m^{i_m}}$ is the activation degree of membership function i_m of the input X_m and Θ is the set of parameters of the fuzzy system. These parameters include the consequents of the rules ($R_{i_1 i_2 \dots i_N}$), the number of membership functions for each input (n_v with $v = 1 \dots N$) and the centers of the membership functions (θ_v^j , where v denotes the input and j is the order of this MF within the set of membership functions, supposing that they are sorted in increasing center order).

The proposed algorithm works in two stages: First, the consequents of the existing rules are adapted with the aim of reducing the plant's output error. When this does not provide any further improvement in the control performance, the topology of the controller is changed by adding a new membership function and creating the corresponding rules (second stage). The consequents of the new rules also need to be adapted, so the algorithm switches back to the first stage. The algorithm does not need any information about the plant's equations or their bounds. Furthermore, it can start working from very simple topologies, even empty ones, which avoids the need of pruning rules. Although its execution never finishes, the second phase is stopped when the desired accuracy level is achieved; this avoids an excessive growth of the number of rules. To measure the control performance and decide when to switch to the second phase, the mean square error (MSE) between the reference signal and the plant output is used [9]. The following sections describe in detail both stages.

2.1 Stage One: Adaptation of the Rule Consequents

In this stage, the controller learns which actions lead to the stabilization of the plant by modifying the consequents of the fuzzy rules. Applying a gradient-descent technique is not feasible, as it requires to compute the partial derivative $\partial y / \partial u$ and our initial hypothesis is that the differential equations that govern the plant are unknown. Nevertheless, as the plant must be controllable, its derivative has a definite constant sign. Hence we can use the information regarding the plant monotonicity with respect to the control input to obtain the right direction in which to move the rule consequents [7,9].

Therefore, our adaptation process consists on the proposal of a correction to the consequents based on the evaluation of the current state of the plant. If the aforementioned monotonicity is positive (i.e. the plant's output grows as the control input grows) and at time $k + 1$ the plant's output is larger than desired ($y(k + 1) > r(k)$), it means that the control signal applied in time k should have been lower; likewise, if $y(k + 1) < r(k)$, the control signal should have been larger. In the case of negative monotonicity, the only difference is that the direction of the changes has to be swapped. Since each rule has a different degree of responsibility on the plant's current state, the penalty applied to each of them is proportional to their degree of activation when obtaining the control input $u(k)$. Therefore, the modification applied to the consequent of the i -th rule at time k is given by:

$$\Delta R_i(k) = C \cdot \mu_i(k-1) \cdot e_y(k) = C \cdot \mu_i(k-1) \cdot (r(k-1) - y(k)) \quad (5)$$

where $\mu_i(k-1)$ is the activation degree of the i -th rule at instant $k-1$, $r(k-1)$ is the set point at that time, $y(k)$ is the current system's output and C is a normalization constant with the same sign as the monotonicity of the plant with respect to u and whose absolute value can be set offline as $|C| = \Delta u / \Delta r$, where Δu is the range of the controller's actuator and Δr is the range in which the reference signal varies (they are both *a priori* known). Note that the expression (5) uses the reference value at instant $k-1$ instead of its current value. The reason for this is that the rules activated at instant $k-1$ served to reach the desired value $r(k-1)$ and not $r(k)$.

On the other hand, most real life controllers have limitations on their operation and this affects the control process. For instance, if the actuator is only able to operate within the range $[u_{min}, u_{max}]$ and at a given moment the optimal control input is $u(k) > u_{max}$, the input finally applied to the plant will be u_{max} , so it will not be possible to reach the desired set point at the next time step. However, we cannot penalize the rules, as they are already giving the best possible answer. To solve this inconvenience, we ensure that no penalty is applied to rules because of the actuator's limitations.

2.2 Stage Two: Modification of the Topology

Most of the adaptive fuzzy controllers proposed at present use fixed structures that are defined beforehand [5]. However, when the plant's dynamics are unknown, choosing the proper topology is not a trivial task [73]. In this case, controllers capable of designing their own structure are needed. To tackle this problem online, information about all the operating regions of the plant needs to be gathered during the controller's normal operation. Most authors [8,10] modify the controller's structure with every new incoming training data, which makes the topology dependable on the sequence of control actions performed. To overcome this, we consider the full operating region; this way, the robustness of the method is increased.

The method proposed here exploits the fact that the very operation of the system provides input/output (I/O) data about the true inverse function of the plant to be controlled. Hence, if the control input $u(k)$ produces the plant's output $y(k+1)$, the output error is not the only information we have. Regardless whether $y(k+1)$ is the desired output or not, we know that if in the same state we find the reference value $r(k') = y(k+1)$ again, the optimal control input will be precisely $u(k)$. With the aim of using this information towards the self-organization of the fuzzy controller, we store the I/O data provided by the plant in a memory M . This information is later used to decide which input needs most a new membership function.

Two steps are performed to modify the fuzzy controller's structure: First, the controller's most relevant input is chosen based on the degree of responsibility that each input has on the approximation error. Then, the new membership function is added and the parameters of the resulting fuzzy controller are initialized with

the aim of minimizing the degradation of the performance right after the change. This two steps are explained next:

Step 1: Selection of the Controller’s Most Relevant Input. The simplest way to modify the controller’s topology is by adding a new MF to every input variable [11]. However, the number of rules depends exponentially on the number of membership functions, so this option is not feasible from a practical point of view [3]. To overcome this problem, it is recommended to choose carefully which variable is going to receive new membership functions. In our case, we base this election on the analysis of the complete error surface reached by the current configuration [12]. Although other methods consider only the point of maximum error, this makes the method more sensible to noise [3].

The main idea behind the election of the “most important” input is to analyze every one of them separately to check their degree of responsibility on the current approximation error. Let us assume we are analyzing the input x_v ; in this case, we assign a large number of membership functions (N_∞) to all the other inputs. This is equivalent to having a perfect approximation in those dimensions and, therefore, only x_v is responsible for the approximation error.

Let F_∞ be the approximation of the data stored in memory M produced by the fuzzy system with the mentioned topology. In this case, we can compute the responsibility index for the input x_v (RI_v) as:

$$RI_v = \sum_{i=1}^K (u^M(\mathbf{x}_i^M) - F_\infty^v(\mathbf{x}_i^M))^2 \quad (6)$$

where K is the number of data stored in memory M , \mathbf{x}_i^M is the i -th input vector stored in memory M , and $u^M(\mathbf{x}_i)$ is the output produced by that input.

Finally, the input variable with the largest responsibility index is the one selected to receive a new membership function: its number of membership functions is increased by one and the functions are equidistributed through all the variable’s range of operation. The addition of this new MF implies the creation of new rules. The second step of this phase handles the initialization of the consequents for the resulting fuzzy controller.

Step 2: Initialization of the Rules of the New Fuzzy Controller. Since we are using a complete set of rules, every time a new MF is added to an input

x_v , $\prod_{\substack{i=1 \\ i \neq v}}^N n_i$ new fuzzy rules are created. Although it is possible to initialize the

new consequents to random values and let them be adapted by the algorithm’s first phase, this would cause a sudden decrease on the control quality in the first moments after the topology modification. To avoid this situation, we initialize the new rules to values that guarantee minimum quality degradation.

Let $\tilde{\Theta}$ be the set of parameters before the new MF was added and Θ the new set of parameters. The idea is that the outline of the global function represented by the fuzzy system is kept the same as before, i.e. $\hat{F}(\mathbf{x}; \Theta) = \hat{F}(\mathbf{x}; \tilde{\Theta}) \forall \mathbf{x}$.

To achieve this, we impose that at the point of maximum activation of the rule, the consequent equals the output produced by the system under its previous configuration for the same input. As the maximum activation degree is reached when all the inputs are located at the centres of the membership functions of the antecedent, we have that:

$$R_{i_1, \dots, i_v, \dots, i_N} = \hat{F}(\mathbf{c}; \tilde{\Theta}) \tag{7}$$

where $\mathbf{c} = (\theta_1^{i_1}, \dots, \theta_v^{i_v}, \dots, \theta_N^{i_N})$, with $i_1 = 1, \dots, n_1, \dots, i_v = 1, \dots, n_v, \dots, i_N = 1, \dots, n_N$ (i.e. all the rules have to update their consequents).

3 Simulation Results

To provide a better insight of the algorithm simulation results are provided in this section. Let us consider the plant commonly known as *water tank* (see Figure 1). This plant represents a tank that has a valve allowing the introduction of liquid, and a vent in its lower side that lets the liquid out. The control objective is to adjust the power of the water entrance so as to achieve a determined height H in the water level. The amount of water that gets in the tank is proportional to the voltage applied to the entrance valve, whilst the water that flows out is proportional to the square root of the level reached by the liquid inside. It is obvious that the amount of water in the tank is equal to the difference between the amount of water getting in and the amount of water getting out. Thus, the differential equation that defines the water level stored in the tank at any time is the following [13]:

$$\frac{dVol}{dt} = A \frac{dH}{dt} = bV - a\sqrt{H} \tag{8}$$

where Vol is the water volume inside the tank, A is the area of the transversal section of the tank, b is a constant associated to the water entrance rate, a is a constant related to the water exit rate and H is the water level inside the tank at a specific moment in time. Despite the formula's simplicity, the presence of a square root makes the differential equation to be non-linear, thus making its analysis difficult. This system also has the peculiarity of giving the actuator a limited range of operation, because negative signals cannot be applied. If the water level has to decrease, the only possible action is to stop letting water in the tank and wait for the liquid to flow out of the tank.

For this example we have used a fuzzy controller with two inputs (the reference signal $r(k)$ and the plant's output $y(k)$). Initially, the controller is empty, which means that every input variable has only one membership function and the only rule is initialized to zero. For the simulation, a reference signal composed by several random step functions within the range $[0.5, 4]$ is used, forming a 1000-iterations long pattern. The sampling rate is 10 samples per second. Parameters for the used plant are: $A = 10$, $a = 1$, $b = 2.5$. The actuator operates within the range $[0, 5]$.

Table 1 shows the evolution of the self-organization process, with each row representing a topology change. For each of them we show the new topology

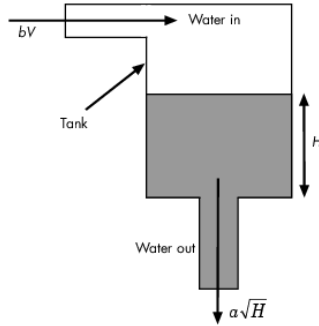


Fig. 1. Water tank system

obtained, the values of the indexes SSE_i used to decide the next change and the mean square error reached for the given topology after the adaptation of the new consequents. Note that the values of the MSE have been multiplied by 10^3 in order to make them more readable. The first two changes set the topology 2×2 , which means that both variables are relevant for the control process. After that, the importance of both inputs is kept similar, as we reach an even topology (4×4). However, in the end, $r(k)$ becomes more important, as it is observed from the fact that it receives six MFs, compared to the four assigned to $y(k)$. This means that, in order to achieve a better accuracy, the control policy has to be finer for the values of $r(k)$.

On the other hand, Figure 2 compares the plant’s output with the desired reference signal along different moments of the execution. Figure 2(a) shows how the empty controller at the beginning of the execution is incapable of controlling the system properly. This is because it does not have any information about the plant and we are starting working with an empty controller. However, after a

Table 1. Controller’s Self-organization Process

| Configuration | SSE_1 | SSE_2 | $MSE \cdot 10^3$ |
|---------------|-----------------------|-----------------------|------------------|
| 1x1 | 253.729 | 237.603 | 710.703 |
| 2x1 | 0.452 | 208.457 | 248.367 |
| 2x2 | $7.104 \cdot 10^{-4}$ | 2.584 | 52.434 |
| 2x3 | $1.989 \cdot 10^{-4}$ | $4.525 \cdot 10^{-4}$ | 40.922 |
| 2x4 | $1.219 \cdot 10^{-4}$ | $8.874 \cdot 10^{-5}$ | 34.033 |
| 3x4 | $9.639 \cdot 10^{-5}$ | $6.152 \cdot 10^{-5}$ | 29.353 |
| 4x4 | $4.624 \cdot 10^{-5}$ | $3.536 \cdot 10^{-5}$ | 26.484 |
| 5x4 | $5.497 \cdot 10^{-5}$ | $2.961 \cdot 10^{-5}$ | 24.809 |
| 6x4 | | | 23.561 |

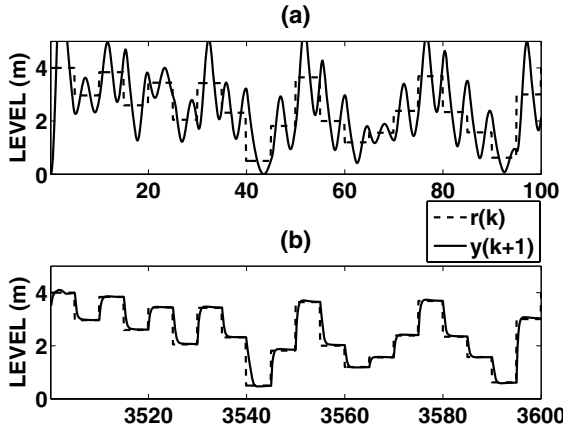


Fig. 2. Tracking of the reference in the water tank system. (a) At the beginning. (b) After one hour.

few minutes the control starts improving, as the algorithm learns, until reaching a very good performance before one hour elapses (Figure 2(b)). Note that the biggest errors in the tracking process are due to the limitation of the actuator: in Figure 2(b) it can be observed that the decrease of the water level is slower than the increase. As mentioned before, this happens because the only possible action for lowering the liquid level is closing the valve.

Finally, we analyze the robustness of the algorithm against unexpected changes in the plant being controlled. The proposed method does not use any specific information or makes any hypotheses about the system it controls, except for the sign of the monotonicity with respect to the control signal u . However it uses the I/O data collected from the system operation itself, and therefore if an

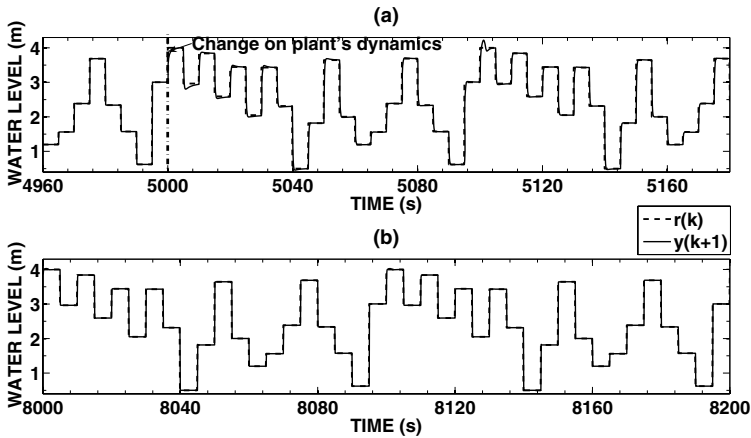


Fig. 3. Reference tracking for the *water tank* plant when the plant's dynamics change

internal change in the plant happens, the information supplied will allow the adaptation of the approximation to the new characteristics.

In order to visualize this property, the value of parameter b in (8) has been reduced during execution. This induces the applied voltage V to produce a lower water entrance rate, which is equivalent to a decay on the actuator's performance due to its use. Initially, plant parameters are $A = 10$, $a = 1$, $b = 10$. At instant 5000, the value of b is reduced to half its initial value. In Figure 3(a) the effect of this change over the tracking of the reference signal can be observed: a clear deterioration in the performance of the control takes place. Nevertheless, the algorithm starts counteracting this deterioration from the beginning to such an extent that after some minutes the control is as good as it was before the change in the plant dynamics (Figure 3(b)).

4 Conclusions

In this work, we have proposed an online adaptive self-organizing fuzzy controller. Without any offline pretraining this system is capable to adapt both the rule consequents and the system topology online, based on I/O data obtained from the plant during the system's normal operation. It is also able to determine which variables need more membership functions and where to locate such functions in order to improve the control performance. The simulations with a mechanic suspension system have shown its capability to perform a high-quality control even if starting from an empty configuration.

Acknowledgments. This work was supported by the Spanish Junta de Andalucía, Consejería de Innovación, Ciencia y Empresa, under Project of Excellence no. TIC02906 and by the Spanish Ministry of Industry, Tourism and Commerce, under project no. TSI-020100-2008-258 (Subprograma Avanza I+D).

References

1. Park, J.-H., Park, G.-T., Kim, S.-H., Moon, C.-J.: Direct adaptive self-structuring fuzzy controller for nonaffine nonlinear system. *Fuzzy Sets and Systems* 153(3), 429–445 (2005)
2. Castro, J.: Fuzzy logic controllers are universal approximators. *IEEE Transactions on Systems, Man and Cybernetics* 25(4), 629–635 (1995)
3. Phan, P.A., Gale, T.J.: Direct adaptive fuzzy control with a self-structuring algorithm. *Fuzzy Sets and Systems* 159(8), 871–899 (2008)
4. Lalouni, S., Rekioua, D., Rekioua, T., Matagne, E.: Fuzzy logic control of stand-alone photovoltaic system with battery storage. *Journal of Power Sources* 193(2), 899–907 (2009)
5. Wang, W., Chien, Y., Li, I.: An on-line robust and adaptive T-S fuzzy-neural controller for more general unknown systems. *International Journal of Fuzzy Systems* 10(1), 33–43 (2008)
6. Mingzhi, H., Jinquan, W., Yongwen, M., Yan, W., Weijiang, L., Xiaofei, S.: Control rules of aeration in a submerged biofilm wastewater treatment process using fuzzy neural networks. *Expert Systems with Applications* 36(7), 10428–10437 (2009)

7. Rojas, I., Pomares, H., Gonzalez, J., Herrera, L., Guillen, A., Rojas, F., Valenzuela, O.: Adaptive fuzzy controller: Application to the control of the temperature of a dynamic room in real time. *Fuzzy Sets and Systems* 157(16), 2241–2258 (2006)
8. Angelov, P.: A fuzzy controller with evolving structure. *Information Sciences* 161(1-2), 21–35 (2004)
9. Pomares, H., Rojas, I., Gonzalez, J., Damas, M., Pino, B., Prieto, A.: Online global learning in direct fuzzy controllers. *IEEE Transactions on Fuzzy Systems* 12(2), 218–229 (2004)
10. Lin, C., Lin, C., Lee, C.S.G.: Fuzzy adaptive learning control network with on-line neural learning. *Fuzzy Sets Syst.* 71(1), 25–45 (1995)
11. Wang, L.: A dynamically generated fuzzy neural network and its application to torsional vibration control of tandem cold rolling mill spindles. *Engineering Applications of Artificial Intelligence* 15(6), 541–550 (2002)
12. Pomares, H., Rojas, I., Gonzalez, J., Prieto, A.: Structure identification in complete rule-based fuzzy systems. *IEEE Transactions on Fuzzy Systems* 10(3), 349–359 (2002)
13. Ogata, K.: *Modern Control Engineering*, 4th edn. Prentice-Hall, Englewood Cliffs (2001)

A Mechanism of Output Constraint Handling for Analytical Fuzzy Controllers

Piotr M. Marusak

Institute of Control and Computation Engineering, Warsaw University of Technology,
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland
P.Marusak@ia.pw.edu.pl

Abstract. In the proposed mechanism the output constraints are handled in a relatively easy way. The method is based on prediction generation known from the MPC (Model Predictive Control) algorithms. It can be, however, used in the case of practically any analytical fuzzy controller. The big advantage of the proposed mechanism is possibility to take into consideration influence of the control action many sampling instants ahead. Therefore, the constraint handling can offer very good control performance.

Keywords: fuzzy control, fuzzy systems, nonlinear control, constrained control, soft computing.

1 Introduction

Output constraints often decide on safety and/or economic efficiency of the process. In the case when the control action must be generated frequently and, therefore, relatively simple analytical controllers (with explicit control law) are used, the constraint handling mechanism must be as simple as possible, though efficient. The method proposed in the paper has these features.

In the proposed method the prediction generation known from the MPC algorithms is used; see e.g. [13, 79]. Then the control action which was generated by a controller is modified in such a way that the predicted output does not violate the constraints. It should be stressed, that, unlike in other methods designed for analytical controllers [5], in the proposed approach the predicted output many sampling instants ahead is taken into consideration during constraint handling. Moreover, the modeling inaccuracy can be easily taken into consideration in the proposed method. Very good performance offered by the method is demonstrated in the example control system of a nonlinear control plant with delay.

2 Mechanism of Output Constraint Handling

It is assumed that the control plant is described by the Takagi–Sugeno model with local models in the form of difference equations:

Rule f : (1)

if y_k is B_1^f and ... and y_{k-n+1} is B_n^f and u_k is C_1^f and ... and u_{k-m+1} is C_m^f
then $y_{k+1}^f = b^{1,f} \cdot y_k + \dots + b^{n,f} \cdot y_{k-n+1} + c^{1,f} \cdot u_k + \dots + c^{m,f} \cdot u_{k-m+1}$,

where $b^{1,f}, \dots, b^{n,f}, c^{1,f}, \dots, c^{m,f}$ are coefficients of the f^{th} local (linear) model, y_k is the value of the output variable at the k^{th} sampling instant, u_k is the manipulated variable value at the k^{th} sampling instant, $B_1^f, \dots, B_n^f, C_1^f, \dots, C_m^f$ are fuzzy sets, $f = 1, \dots, l$, l is the number of fuzzy rules.

In general, output of the fuzzy model is described by the following formula:

$$\hat{y}_{k+1} = \sum_{j=1}^n \tilde{b}_k^j \cdot y_{k-j+1} + \sum_{j=1}^m \tilde{c}_k^j \cdot u_{k-j+1} \quad , \quad (2)$$

where $\tilde{b}_k^j = \sum_{f=1}^l \tilde{w}_k^f \cdot b^{j,f}$, $\tilde{c}_k^j = \sum_{f=1}^l \tilde{w}_k^f \cdot c^{j,f}$, \tilde{w}_k^f are normalized weights obtained using the fuzzy reasoning (see e.g. [6][8]); they depend, in general, on past output and control values, however only their dependence on time is denoted for brevity and clarity of description.

Assuming that the control action will not change ($u_k = u_{k+1} = \dots$), behavior of the control plant can be predicted many sampling instants ahead using iterative method, i.e. iteratively using the fuzzy model (2). Thus, the predicted output values are described by the following formula:

$$\hat{y}_{k+i} = \sum_{j=1}^{i-1} \tilde{b}_{k+i}^j \cdot \hat{y}_{k-j+i} + \sum_{j=i}^n \tilde{b}_{k+i}^j \cdot y_{k-j+i} + \sum_{j=1}^i \tilde{c}_{k+i}^j \cdot u_k + \sum_{j=i+1}^m \tilde{c}_{k+i}^j \cdot u_{k-j+i} \quad , \quad (3)$$

where \hat{y}_{k+i} is the output of the fuzzy model for the $(k+i)^{\text{th}}$ sampling instant.

Moreover, it is advisable to assess and take into consideration the modeling inaccuracy and influence of unmeasured disturbances. If one does not have the disturbance and modeling error estimates, one can adapt a mechanism used in the MPC algorithms. It consists in calculating difference between measured output and output of the model

$$d_k = y_k - \hat{y}_k \quad . \quad (4)$$

Then it is assumed that d_k will be the same in the next sampling instants. Therefore, finally, one obtains the following prediction:

$$y_{k+i|k} = \sum_{j=1}^i \tilde{c}_{k+i}^j \cdot u_k + \sum_{j=i+1}^m \tilde{c}_{k+i}^j \cdot u_{k-j+i} + \sum_{j=1}^{i-1} \tilde{b}_{k+i}^j \cdot \hat{y}_{k-j+i} + \sum_{j=i}^n \tilde{b}_{k+i}^j \cdot y_{k-j+i} + d_k \quad . \quad (5)$$

The prediction (5) can be written in more compact form as:

$$y_{k+i|k} = C_{k+i} \cdot u_k + D_{k+i} \quad , \quad (6)$$

where $D_{k+i} = \sum_{j=i+1}^m \tilde{c}_{k+i}^j \cdot u_{k-j+i} + \sum_{j=1}^{i-1} \tilde{b}_{k+i}^j \cdot \hat{y}_{k-j+i} + \sum_{j=i}^n \tilde{b}_{k+i}^j \cdot y_{k-j+i} + d_k$,
 $C_{k+i} = \sum_{j=1}^i \tilde{c}_{k+i}^j$.

Usually, one demands that output value does not violate the limits which do not change in time. Therefore, the output constraints which should be fulfilled are as follows:

$$y_{\min} \leq y_{k+i|k} \leq y_{\max} \quad , \tag{7}$$

where y_{\min} and y_{\max} are lower and upper output limits, respectively. Thanks to using the prediction (6) the output constraints can be applied many samplings instants ahead:

$$y_{\min} \leq C_{k+i} \cdot u_k + D_{k+i} \leq y_{\max} \quad . \tag{8}$$

The output constraints (8) can be transformed into sets of constraints put on the currently derived control value for lower constraints:

$$C_{k+i} \cdot u_k \geq y_{\min} - D_{k+i} \tag{9}$$

and for upper constraints:

$$C_{k+i} \cdot u_k \leq y_{\max} - D_{k+i} \quad . \tag{10}$$

Next, the following rules of control value modification should be applied:

- for lower constraints:
 - if $C_{k+i} \cdot u_k \geq y_{\min} - D_{k+i}$ then $u_k = \frac{y_{\min} - D_{k+i}}{C_{k+i}}$ and
- for upper constraints:
 - if $C_{k+i} \cdot u_k \leq y_{\max} - D_{k+i}$ then $u_k = \frac{y_{\max} - D_{k+i}}{C_{k+i}}$.

Remark 1. The key issue is to use the modified control values (actually applied to the plant) in the next iterations during control signal calculation by the controller. Otherwise, control performance may be degraded.

Remark 2. The parameters \tilde{b}_{k+i}^j and \tilde{c}_{k+i}^j may depend, in general, on u_k and future output values. Therefore, the method above is an approximate one. However, if a problem needs improvement of constraint satisfaction, the modification of control value u_k may be repeated a few times, iteratively, in order to correct the result.

Remark 3. The calculations in the proposed mechanism are not too complicated. Moreover, one can decide how many sampling instants ahead output of the control plant will be predicted and constrained. Thus, the mechanism of constraint handling may be easily tailored to the particular problem.

Remark 4. Analytical controllers equipped with the proposed mechanism can be applied in constrained control systems for which other types of controllers (e.g. numerical MPC algorithms based on optimization problem solved at each iteration of the algorithm, often used in constrained control systems) are too complex and too time consuming.

In the proposed method a basic mechanism of modeling inaccuracy assessment known from MPC algorithms may be used, as discussed earlier. However, if one can assess the values of modeling errors, this assessment can be used to introduce

a safety margin to the output constraints. The better the assessment of the modeling error, the better results will be obtained.

The output prediction with uncertainty can be described by:

$$\check{y}_{k+i|k} = y_{k+i|k} + r_{k+i|k} = C_{k+i} \cdot u_k + D_{k+i} + r_{k+i|k} \quad , \quad (11)$$

where components $r_{k+i|k}$ represent influence of the modeling error on the prediction; they are usually unknown. If the following assessment of the minimum and maximum values of the $r_{k+i|k}$ was done:

$$r_{k+i|k}^{\min} \leq r_{k+i|k} \leq r_{k+i|k}^{\max} \quad , \quad (12)$$

where $r_{k+i|k}^{\min} \leq 0$ and $r_{k+i|k}^{\max} \geq 0$, then the following rules of control value modification should be applied:

- for lower constraints:
 - if $C_{k+i} \cdot u_k \geq y_{\min} - D_{k+i} - r_{k+i|k}^{\min}$ then $u_k = \frac{y_{\min} - D_{k+i} - r_{k+i|k}^{\min}}{C_{k+i}}$ and
- for upper constraints:
 - if $C_{k+i} \cdot u_k \leq y_{\max} - D_{k+i} - r_{k+i|k}^{\max}$ then $u_k = \frac{y_{\max} - D_{k+i} - r_{k+i|k}^{\max}}{C_{k+i}}$.

Remark 5. In practice, the further in the future a predicted value of the output variable is, the more difficult (and more conservative) the assessment of the modeling error usually is. Therefore, in practice, one can apply the control value modification rules with the same modeling error assessments for all future sampling instants ($r_{k+i|k}^{\min} = r_{\min}$ and $r_{k+i|k}^{\max} = r_{\max}$) where r_{\min} and r_{\max} may be equal, e.g. to the values obtained for the $(k + 1)^{\text{st}}$ sampling instant, i.e. $r_{\min} = r_{k+1|k}^{\min}$ and $r_{\max} = r_{k+1|k}^{\max}$. The other solution is to resign from taking the modeling uncertainty into consideration in the further, than assumed, sampling instants. The prediction and constraint handling mechanism can be applied in the next iteration of the controller, with updated modeling inaccuracy assessment, anyway.

3 Simulation Experiments

The proposed mechanism is tested in the control system of a distillation column – a nonlinear plant with delay. The control plant is described by the fuzzy Takagi–Sugeno model which consists of three following rules (the sampling time $T_s = 40$ min was assumed):

Rule f : if u_{k-2} is M_f , then

$$y_{k+1}^f = b^f \cdot y_k + c^f \cdot u_{k-2} + d^f \quad , \quad (13)$$

where $f = 1, 2, 3$ is the index of the fuzzy rules, $b^1 = b^2 = b^3 = 0.7659$, $c^1 = -520.2638$, $c^2 = -253.5771$, $c^3 = -125.1030$, $d^1 = 2220.9067$, $d^2 = 1102.4471$, $d^3 = 563.8767$. The membership functions are shown in Fig. [11](#).

The output variable y is the impurity of the product (counted in ppm). It is assumed that the output is constrained $y \leq 400$ ppm due to quality demand.

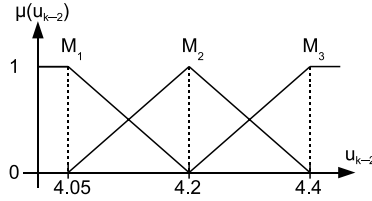


Fig. 1. Membership functions of the control plant model

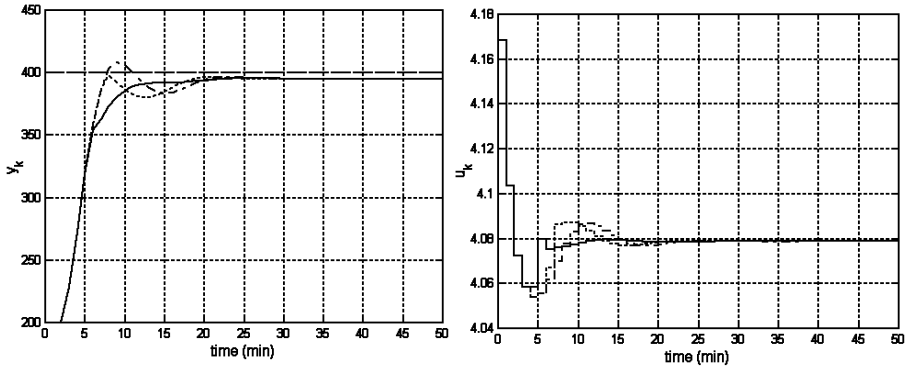


Fig. 2. Responses of the control system with FDMC controller to the change of the set-point value to $\bar{y} = 395$ ppm; output constraints: not taken into consideration (dash-dotted lines), taken into consideration in only one next sampling instant (dotted lines) and taken into consideration in next three sampling instants (solid lines); dashed line – constraint

The allowable impurity cannot be exceeded, otherwise the product will be wasted. The manipulated variable u is the reflux to product ratio (the higher it is the purer product is obtained).

The fuzzy model was used to design an analytical fuzzy DMC (FDMC) controller (see e.g. [4,9]) and in the constraint handling mechanism. In order to test the proposed approach in presence of modeling inaccuracy, the second model, of Hammerstein structure [2] with polynomial model of the statics, served as the control plant during simulation experiments. Only the basic mechanism of modeling inaccuracy assessment was used.

The example responses are shown in Fig. 2. The set-point value is set to $\bar{y} = 395$ ppm. If the mechanism of output constraint handling is not applied then the constraint is violated (dash-dotted lines in Fig. 2). Application of the approach in which only one (next) predicted output value is constrained brings fulfillment of the constraint but the output is oscillating and the response is not smooth (dotted lines in Fig. 2). Application of the proposed approach for next three predicted output values gives the best result (solid lines in Fig. 2).

The constraint is fulfilled and the output achieves the set-point value fast and practically without overshoot.

Usage of the proposed mechanism improved operation of the controller and made the output signal smoother than in the case when only one predicted output value (for the next sampling instant) was taken into consideration. The reason of that can be observed in the control signal. The controller began to take the constraint into consideration (and change the control signal comparing to the case without any constraint handling mechanism) earlier than in the case when only one predicted output value was constrained.

4 Summary

The effective and easy to use mechanism of output constraint handling is proposed in the paper. Thanks to the usage of process behavior prediction, known from the MPC approach, the constraints many sampling instants ahead can be taken into consideration. Therefore, the control signal can be modified in advance what can ensure fulfillment of constraints and improve performance of the control system.

The proposed method, though based on the MPC approach, can be applied to any analytical controller. Moreover, it can be easily adapted to the case when Takagi–Sugeno fuzzy models other than considered in the paper are used, e.g. with state–space equations or step responses utilized as the local models.

Acknowledgment. This work was supported by the Polish national budget funds for science 2009–2011.

References

1. Camacho, E.F., Bordons, C.: *Model Predictive Control*. Springer, Heidelberg (1999)
2. Janczak, A.: *Identification of nonlinear systems using neural networks and polynomial models: a block-oriented approach*. Springer, Heidelberg (2005)
3. Maciejowski, J.M.: *Predictive control with constraints*. Prentice Hall, Harlow (2002)
4. Marusak, P.: Machine tuning of stable analytical fuzzy predictive controllers. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) *ICANNGA 2009*. LNCS, vol. 5495, pp. 430–439. Springer, Heidelberg (2009)
5. Marusak, P., Tatjewski, P.: Output constraints in fuzzy DMC algorithms with parametric uncertainty in process models. In: *Proc. 7th Int. Conf. on Methods and Models in Automation and Robotics MMAR 2001*, Miedzyzdroje, Poland, pp. 517–522 (2001)
6. Piegat, A.: *Fuzzy Modeling and Control*. Physica–Verlag, Berlin (2001)
7. Rossiter, J.A.: *Model-Based Predictive Control*. CRC Press, Boca Raton (2003)
8. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. Systems, Man and Cybernetics* 15, 116–132 (1985)
9. Tatjewski, P.: *Advanced Control of Industrial Processes; Structures and Algorithms*. Springer, London (2007)

Analysis of the Performance of a Semantic Interpretability-Based Tuning and Rule Selection of Fuzzy Rule-Based Systems by Means of a Multi-Objective Evolutionary Algorithm*

María José Gacto, Rafael Alcalá, and Francisco Herrera

Dept. Computer Science, University of Jaén, Jaén, Spain
Dept. Computer Science and A.I., University of Granada, Granada, Spain
mjgacto@ugr.es, {alcala,herrera}@decsai.ugr.es

Abstract. Recently, a semantic interpretability index has been proposed to preserve the semantic interpretability of Fuzzy Rule-Based Systems while a tuning of the membership functions is performed. In this work, we extend the proposed multi-objective evolutionary algorithm in order to analyze the performance of the tuning based on this semantic interpretability index while it is combined with a rule selection. To this end, the following three objectives have been considered: error and complexity minimization, and semantic interpretability maximization.

The analyzed method is compared to a single objective algorithm and to the previous approach in two problems showing that many solutions in the Pareto front dominate to those obtained by these methods.

Keywords: Fuzzy Rule-Based Systems, Rule Selection, Tuning, Semantic Interpretability Index, Multi-Objective Evolutionary Algorithms.

1 Introduction

Fuzzy modeling usually tries to improve the accuracy of the system without inclusion of any interpretability measure, an essential aspect of Fuzzy Rule-Based Systems (FRBSs). However, the problem of finding the right trade-off between accuracy and interpretability has achieved a growing interest [1].

Many authors improve the trade-off between accuracy and interpretability of FRBSs, obtaining linguistic models not only accurate also interpretable. We can distinguish two kinds of approaches for managing the interpretability:

1. Measuring the complexity of the model [2,3,4] (usually measured as number of rules, variables, labels per rule, etc.).
2. Measuring the interpretability of the fuzzy partitions [3,5,6,7] by means of a semantic interpretability measure.

* Supported by the Spanish Ministry of Education and Science under grant no. TIN2008-06681-C06-01.

Focusing on the second type, a semantic interpretability index has been proposed in [7] for preserving the interpretability while a *tuning* of the Membership Functions (MFs) is performed. The tuning of MFs enhances the performance of FRBSs and consists of refining the parameters that identify the MFs associated to the labels comprising the Data Base (DB) [2,8,9]. The proposed index is used as an additional measure that quantify the interpretability of the tuned DB. While in the previous works constraints [6] or absolute measures [5] are used, in [7] a relative interpretability index allows to maintain the interpretability of the original MFs that could be given by an expert, by means of the aggregation of several metrics while a tuning is performed.

In this work, we analyze the performance of the combination of the tuning based on the semantic interpretability index together with a rule selection method to reduce the model complexity. The application of Multi-Objective Evolutionary Algorithms (MOEAs) [10,11] allows to obtain a set of solutions with different degrees of accuracy and interpretability [2,4,5,7]. We use an extension of the MOEA proposed in [7] in order to perform a rule selection together with the tuning of MFs with the following three objectives:

- *maximization of the semantic interpretability index*
- *minimization of the number of rules*
- *minimization of the system error*

This algorithm is based on *SPEA2* [12] and is called *TS_{SP2-SI}* (*Tuning and Selection by SPEA2 for Semantic Interpretability*). The combination of the tuning and the rule selection in the same process allows to obtain precise models with an important reduction in the number of rules [13]. In order to analyze this method, it is compared to a single objective accuracy-guided algorithm [13] for tuning and rule selection and to the tuning based on the semantic index in [7]. Two real-world problems have been considered showing that the solutions of the previous approaches are dominated by those obtained by *TS_{SP2-SI}*.

In order to do that, section 2 presents the index to measure the semantic interpretability. Section 3 presents the *TS_{SP2-SI}* algorithm for the tuning of MFs and rule selection. Section 4 analyzes the combined action of rule selection and tuning in terms of the accuracy, complexity and semantic interpretability of the obtained models. Finally, section 5 points out some conclusions.

2 A Semantic Interpretability Index

In this section, we describe several metrics that are combined in a index to measure the interpretability when a tuning is performed on the DB. These metrics were proposed in [7], except one of them that will be generalized here. In order to ensure the semantics integrity through the MFs optimization process [16], some researchers have proposed several properties. Considering one or more of these properties several constraints can be applied in the design process in order to obtain a DB maintaining the linguistic model comprehensibility to the higher possible level [14,15].

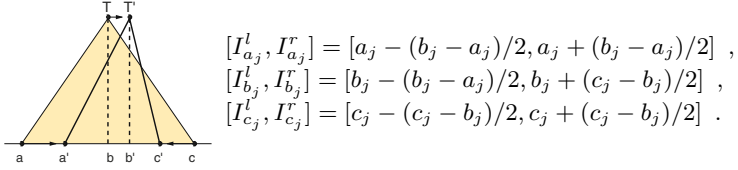


Fig. 1. Tuning by changing the basic MF parameters and Variation intervals

In order to maintain the semantic integrity we consider also these constraints by defining the variation intervals for each MF parameter. Even though we could consider other types of fuzzy partitions, in this work, we use strong fuzzy partitions with triangular MFs defined by means of three parameters (See Figure 1). For each $MF_j = (a_j, b_j, c_j)$ where $j=(1, \dots, m)$ and m is the number of MFs in a given DB, the variation intervals are calculated in the following way:

The metrics in [7] are based on the existence of these variation intervals (integrity constraints). These metrics allow to measure certain characteristics of tuned MFs regarding the original ones. The index and metrics have been proposed for triangular MFs, but they can be easily extended with some small changes in the formulation to Gaussian or trapezoidal. In this work, the metric γ is a generalization that transforms it into a relative metric. These metrics are:

- MFs displacement (δ): This metric measures the proximity of the central points of the MFs to the original ones.
- MFs lateral amplitude rate (γ): This metric measures the left/right rate differences of the tuned and the original MFs.
- MFs area similarity (ρ): This metric measures the area similarity of the tuned MFs and the original ones.

2.1 MFs Displacement Measure (δ)

This metric can control the displacements in the central point of the MFs. It is based on computing the normalized distance between the central point of the tuned MF and the central point of the original MF, and it is calculated through obtaining the maximum displacement obtained on all the MFs. For each MF_j in the DB, we define $\delta_j = |b_j - b'_j|/I$, where $I = (I_{b_j}^r - I_{b_j}^l)/2$ represents the maximum variation for each central parameter. Thus δ^* is defined as $\delta^* = \max_j\{\delta_j\}$. The δ^* metric takes values between 0 and 1, therefore values near to 1 show that the MFs present a great displacement. The following transformation is made so that this metric represents proximity (maximization):

$$\text{Maximize } \delta = 1 - \delta^* . \tag{1}$$

2.2 MFs Lateral Amplitude Rate Measure (γ)

This metric can be used to control the MF shapes. It is based on relating the left and right parts of the support of the original and the tuned MFs. Let us define

$leftS_j = |a_j - b_j|$ as the amplitude of the left part of the original MF support and $rightS_j = |b_j - c_j|$ as the right part amplitude. Let us define $leftS'_j = |a'_j - b'_j|$ and $rightS'_j = |b'_j - c'_j|$ as the corresponding parts in the tuned MFs. γ_j is calculated using the following equation for each MF:

$$\gamma_j = \frac{\min\{leftS_j/rightS_j, leftS'_j/rightS'_j\}}{\max\{leftS_j/rightS_j, leftS'_j/rightS'_j\}} . \tag{2}$$

Values near to 1 mean that the left and right rate in the original MFs are highly maintained in the tuned MFs. Finally γ is calculated by obtaining the minimum value of γ_j :

$$\text{Maximize } \gamma = \min_j\{\gamma_j\} . \tag{3}$$

2.3 MFs Area Similarity Measure (ρ)

This metric can be used to control the area of the MF shapes. It is based on relating the areas of the original and the tuned MFs. Let us define A_j as the area of the triangle representing the original MF_j , and A'_j as the new area. ρ_j is calculated using the following equation for each MF:

$$\rho_j = \frac{\min\{A_j, A'_j\}}{\max\{A_j, A'_j\}} . \tag{4}$$

Values near to 1 mean that the original area and the tuned area of the MFs are more similar (less changes). The ρ metric is calculated by obtaining the minimum value of ρ_j :

$$\text{Maximize } \rho = \min_j\{\rho_j\} . \tag{5}$$

2.4 Semantics Based Interpretability Index: GM3M

The semantic interpretability index, namely GM3M proposed in [7], is defined as the geometric mean of the three metrics. The geometric mean is used because in case that only one of the metrics has very low values (causing low interpretability) it is also obtained small values of GM3M. The index is defined as:

$$\text{Maximize } GM3M = \sqrt[3]{\delta \cdot \gamma \cdot \rho} \tag{6}$$

The value of GM3M ranges between 0 (the lowest level of interpretability) and 1 (the highest level of interpretability).

3 MOEA for Rule Selection and Tuning of FRBSs

The presented algorithm is an extension of the MOEA proposed in [7], to perform a tuning based on the semantic interpretability index $GM3M$, while it is combined with a rule selection. It is called Tuning and Selection by SPEA2 for Semantic Interpretability (TS_{SP2-SI}) and is based on the well-known SPEA2 [12] algorithm. TS_{SP2-SI} implements such concepts as incest prevention and restarting [16], and incorporates the main ideas of the algorithm proposed in [2] for

guiding the search towards the desired Pareto zone. Thus, the presented algorithm is aimed at generating a complete set of Pareto-optimum solutions with different trade-offs between accuracy and interpretability. We have chosen as base of our method the SPEA2 algorithm since in [2], approaches based on SPEA2 were shown to be more effective when performing a tuning of the MFs. In the next subsections the main components of this algorithm are described and the specific characteristics are presented.

3.1 Coding Scheme and Initial Gene Pool

A double coding scheme for both *rule selection* (C_S) and *tuning* (C_T) is used: $C^p = C_S^p C_T^p$. In the $C_S^p = (c_{S1}, \dots, c_{Sm})$ part, the coding scheme consists of binary-coded strings with m being the number of initial rules. Depending on whether a rule is selected or not, values ‘1’ or ‘0’ are respectively assigned to the corresponding gene. In the C_T part a real coding is used, being m^i the number of labels of each of the n variables in the DB,

$$C_T^p = C_1 C_2 \dots C_n; C_i = (a_1^i, b_1^i, c_1^i, \dots, a_{m^i}^i, b_{m^i}^i, c_{m^i}^i), i = 1, \dots, n .$$

The initial population is obtained with all individuals having all genes with value ‘1’ in C_S . In the C_T part, the initial DB is included as a first individual and the remaining individuals are generated at random within the corresponding variation intervals defined in previous section.

3.2 Objectives

The objectives considered in this algorithm are:

1. Semantic interpretability maximization: Semantic based index (GM3M).
2. Complexity minimization: Number of Rules (NR).
3. Error minimization: Mean Squared Error (MSE).

$MSE = \frac{1}{2 \cdot |E|} \sum_{l=1}^{|E|} (F(x^l) - y^l)^2$, where $|E|$ is the dataset size, $F(x^l)$ is the output of the FRBS when the l -th example is an input and y^l is the known desired output. The fuzzy inference system uses the *center of gravity weighted by the matching* strategy as a defuzzification operator and the *minimum t-norm* as implication and conjunctive operators.

3.3 Crossover and Mutation

This method uses an intelligent crossover and a mutation operator that it allows to take advantage of the information contained in the parents, improving the balance between exploration and exploitation. Due to the method uses different codes in each one of the parts of the chromosome, it is necessary to apply different operators in each area. The steps to obtain each offspring are as follows:

- BLX-0.5 [17] crossover is applied to obtain the C_T part of the offspring.
- Once the offspring C_T part has been obtained, the binary part C_S is obtained based on the C_T parts (MFs) of parents and offspring. For each gene in the C_S part which represents a concrete rule:
 1. The MFs involved in such rule are extracted from the corresponding C_T parts for each individual involved in the crossover (offspring and parents 1 and 2). Thus, we can obtain the specific rules that each of the three individuals are representing.
 2. Euclidean normalized distances are computed between the offspring rule and each parent rule by considering the center points (vertex) of the MFs comprising such rules.
 3. The parent with the closer rule to the one obtained by the offspring is the one that determines if this rule is selected or not for the offspring by directly copying its value in C_S for the corresponding gene.

This process is repeated until all the C_S values are assigned for the offspring. By applying this operator, exploration is performed in the C_T part, and C_S is directly obtained based on the previous knowledge of each parent. The **mutation operator** does not need to add rules, directly sets to zero a gene selected at random in the C_S part. In the C_T part changes a gene value at random. Four offspring are obtained repeating this process four times, only the two most accurate are taken as descendants.

3.4 Main Characteristics of TS_{SP2-SI}

This algorithm makes use of the *SPEA2* selection mechanism. However in order to improve its search ability the following changes are considered:

- It includes a mechanism for incest prevention based on the concepts of CHC [16], for maintaining population diversity. This mechanism avoids premature convergence. Only those parents whose hamming distance divided by 4 is higher than a threshold are crossed. Since we consider a real coding scheme in C_T , we have to transform each gene using a Gray Code with a fixed number of bits per gene (*BGene*). In this way, the threshold value is initialized as $L = (\#C_T * BGene)/4$, where $\#C_T$ is the number of genes in the C_T part of the chromosome. At each generation of TS_{SP2-SI} , the threshold value is decremented by one which allows crossing over closer solutions.
- A restarting operator is applied by maintaining the most accurate individual, and the most interpretable individual as a part of the new population. The remaining individuals in the new population take the values of the most accurate individual in the C_S part and values generated at random in the C_T part. We apply the first restart if 50 percent of crossovers are detected at any generation (this percentage is updated each time restarting is performed as $\%_r = (1 + \%_r)/2$). Moreover, the most accurate solution should be improved before each restarting. To preserve a well formed Pareto front, the restarting is not applied in the last evaluations of the algorithm (i.e., when the number of evaluations is equal to the number of evaluations consumed until the first restart multiplied by 10)

- In each stage of the algorithm (between restarting points), the number of solutions in the external population (\overline{P}_{t+1}) considered to form the mating pool is progressively reduced, by focusing only on those with the best accuracy. To do that, the solutions are sorted from the best to the worst (considering accuracy as sorting criterion) and the number of solutions considered for selection is reduced progressively from 100% at the beginning to 50% at the end of each stage by taking into account the value of L . In the last evaluations this mechanism, whose main objective is focusing on the most accurate solutions, is also disabled in order to obtain a wide well formed Pareto front.

4 Analysis of the Performance of the Combined Action of Both, Rule Selection and Tuning

To analyze the performance of the tuning based on the semantic interpretability index together with a rule selection, two real-world regression problems with different complexities (different number of variables and available data) are considered to be solved (these data sets are available at, <http://www.keel.es/>):

1. Predicting the Weather in Izmir (WIZ): 10 variables and 1461 examples.
2. Predicting the Mortgage Rate (MOR): 16 variables and 1049 examples.

The methods considered for the experiments are:

- TS which performs the rule selection and a tuning of the MFs by only considering the accuracy of the model as the sole objective [13].
- T_{SP2-SI} , the MOEA applying tuning with GM3M as a second objective [7].
- $T_{SSP2-SI}$ is the presented MOEA for the combination of rule selection and the tuning considering the three objectives mentioned previously.

The Wang and Mendel method (WM) [18] is used to obtain the initial Rule Bases (RBs) that will be tuned by the said methods. We consider a *5-fold cross-validation model*, i.e., 5 random partitions of data each with 20%, and the combination of 4 of them (80%) as training and the remaining one as test. For each one of the 5 data partitions, the considered methods have been run 6 times, showing for each problem the averaged results of a total of 30 runs. In the case of $T_{SSP2-SI}$ and T_{SP2-SI} the averaged values are calculated considering the most accurate solution from each Pareto front obtained.

The values of the input parameters considered by TS are: population size of 61, 100000 evaluations, 0.6 as crossover probability and 0.2 as mutation probability per chromosome. The values of the input parameters considered by the MOEAs are: population size of 200, external population size of 61, 100000 evaluations, 0.2 as mutation probability and 30 bits per gene for the Gray codification.

Table 1 shows the results obtained with WM, where NR stands for the number of rules, $MSE_{tra/tst}$ for the averaged error obtained over the training/test data, σ for their respective standard deviations and $GM3M$ for the semantic interpretability index. In the WM method, $GM3M$ takes value 1 that is the highest level of interpretability since the MFs are not modified.

Table 1. Results obtained with WM method

| Dataset | NR | MSE _{tra} | σ_{tra} | MSE _{tst} | σ_{tst} | GM3M |
|---------|-------|--------------------|----------------|--------------------|----------------|------|
| WIZ | 104.8 | 6.944 | 7.368 | 0.720 | 0.909 | 1 |
| MOR | 77.6 | 0.985 | 0.973 | 0.129 | 0.090 | 1 |

Table 2. Results obtained in both problems

| Dataset | Method | Ref | NR | MSE _{tra} | σ_{tra} | t-test | MSE _{tst} | σ_{tst} | t-test | GM3M | σ_{GM3M} | t-test | (δ | γ | ρ) | |
|---------|-----------------|------|-------------|--------------------|----------------|--------|--------------------|----------------|--------|--------------|-----------------|--------|------------|----------|----------|-------|
| WIZ | <i>TS</i> | [13] | 53.5 | 1.051 | 0.07 | + | 2.386 | 1.95 | + | 0.349 | 0.08 | + | (0.16 | 0.45 | 0.71) | |
| | <i>TSP2-SI</i> | [4] | 104.8 | 1.048 | 0.04 | + | 1.243 | 0.12 | + | 0.260 | 0.11 | + | (| 0.10 | 0.41 | 0.66) |
| | <i>TSSP2-SI</i> | - | 29.2 | 0.921 | 0.06 | * | 1.095 | 0.17 | * | 0.493 | 0.15 | * | (0.34 | 0.57 | 0.70) | |
| MOR | <i>TS</i> | [13] | 34.1 | 0.031 | 0.01 | = | 0.037 | 0.01 | = | 0.316 | 0.09 | + | (0.14 | 0.36 | 0.76) | |
| | <i>TSP2-SI</i> | [4] | 77.6 | 0.036 | 0.01 | + | 0.043 | 0.01 | + | 0.183 | 0.07 | + | (0.06 | 0.17 | 0.72) | |
| | <i>TSSP2-SI</i> | - | 15.4 | 0.028 | 0.01 | * | 0.034 | 0.01 | * | 0.541 | 0.10 | * | (0.44 | 0.50 | 0.76) | |

The results obtained by the three analyzed methods are shown in Table 2. In addition, we also show δ , γ and ρ that represent the values of the metrics, and t represents the results of applying a *test t-student* (with 95 percent confidence) in order to ascertain whether differences in the performance of the best results are significant when compared with that of the other algorithm in the table. The interpretation of the t column is: [\star] represents the best averaged result and [$+$] means that the best result has better performance than that of the related row.

Analysing the results showed in Table 2, we can highlight the following facts:

- The studied method obtains the best results in training and test with respect to the other methods in both problems.
- The most accurate solutions from *TSSP2-SI* improve the accuracy and obtain more interpretable models, with 29%(WIZ) and 41%(MOR) of improvement in GM3M with respect to the *TS* method and with 47%(WIZ) and 66%(MOR) of improvement in GM3M with respect to the *TSP2-SI* method.
- *TSSP2-SI* has obtained RBs with almost a half of the rules obtained by *TS*. A great number of rules has been eliminated (more than 60 rules) with respect to the initial RBs obtained with *WM* and with respect to the tuned DB obtained in *TSP2-SI*.

Figure 2 shows the Pareto front obtained with *TSSP2-SI* in MOR, together with the projections in the MSE-Rules, MSE-GM3M and Rules-GM3M planes, and the solution obtained by *TS* and *TSP2-SI* in the corresponding planes. The solutions obtained with *TSP2-SI* and *TS* are dominated by several solutions from *TSSP2-SI*. Moreover, the obtained Pareto front is quite wide and allows selecting solutions with different degrees of accuracy and interpretability.

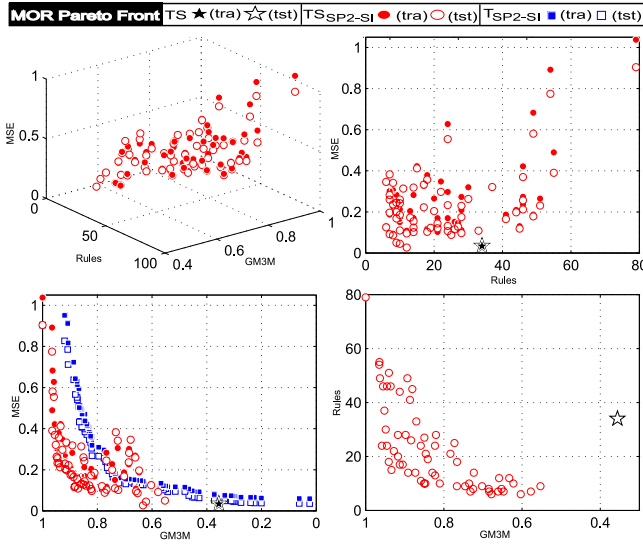


Fig. 2. Pareto Front obtained with the results of a single trial in MOR

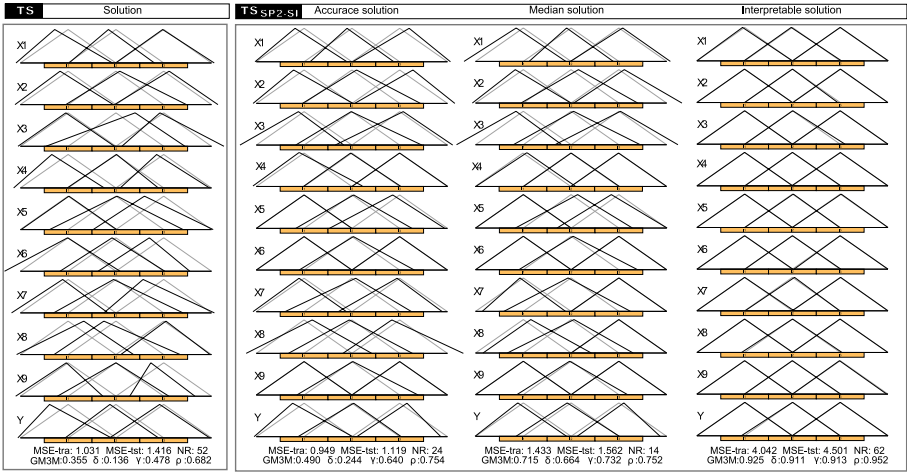


Fig. 3. DBs obtained by TS and TS_{SP2-SI} from same data partition and seed in WIZ

Figure 3 presents a DB obtained with TS and some DBs obtained with the presented method in WIZ. For TS_{SP2-SI} it includes three DBs, one with the most accurate solution, other with a solution not only accurate also interpretable and another highly interpretable DB, that obtains 40% of improvement with respect to the WM method with a value of GM3M near to 1.

5 Concluding Remarks

This work analyzes the performance of the tuning based on semantic interpretability while it is combined with rule selection. The interaction between rule selection and tuning approach with the GM3M index, allows an important reduction of the system complexity and obtain more interpretable and, at the same time, more accurate models, improving the MSE-GM3M trade-off.

The presented method obtains wide well formed Pareto fronts that provide a large variety of solutions to select from more accurate solutions to more interpretable ones. The solutions obtained by the MOEA dominate in general the ones obtained by the mono-objective method and by the previous approach.

References

1. Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (eds.): Interpretability issues in fuzzy modeling. *Studies in Fuzz. and Soft Comp.*, vol. 128. Springer, Heidelberg (2003)
2. Alcalá, R., Gacto, M.J., Herrera, F., Alcalá-Fdez, J.: A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 15(5), 539–557 (2007)
3. Alonso, J.M., Magdalena, L., Guillaume, S.: Hilk: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism. *International Journal of Intelligent Systems* 23(7), 761–794 (2008)
4. Ishibuchi, H., Nojima, Y.: Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning* 44(1), 4–31 (2007)
5. Botta, A., Lazzarini, B., Marcelloni, F., Stefanescu, D.C.: Context adaptation of fuzzy systems through a multi-objective evolutionary approach based on a novel interpretability index. *Soft Computing* 13(5), 437–449 (2008)
6. de Oliveira, J.V.: Towards neuro-linguistic modeling: constraints for optimization of membership functions. *Fuzzy Sets and Systems* 106(3), 357–380 (1999)
7. Gacto, M.J., Alcalá, R., Herrera, F.: A multiobjective evolutionary algorithm for tuning fuzzy rule based systems with measures for preserving interpretability. In: *Proc. of the Joint IFSA World Congress and EUSFLAT Conference*, Lisbon, Portugal, pp. 1146–1151 (2009)
8. Herrera, F., Lozano, M., Verdegay, J.L.: Tuning fuzzy logic controllers by genetic algorithms. *International J. of Approximate Reasoning* 12, 299–315 (1995)
9. Karr, C.: Genetic algorithms for fuzzy controllers. *AI Expert* 6(2), 26–33 (1991)
10. Coello, C.A., Veldhuizen, D.A.V., Lamont, G.B. (eds.): *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Publishers, Dordrecht (2002)
11. Deb, K.: *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, NY (2001)
12. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: *Proc. Evolutionary Methods for Design, Optimization and Control with App. to Industrial Problems*, Barcelona, Spain, pp. 95–100 (2001)

13. Casillas, J., Cordón, O., Jesus, M.J.D., Herrera, F.: Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *IEEE Trans. Fuzzy Syst.* 13(1), 13–29 (2005)
14. Bodenhofer, U., Bauer, P.: A formal model of interpretability of linguistic variables. In: [1], pp. 524–545
15. Espinosa, J., Vandewalle, J.: Constructing fuzzy models with linguistic integrity from numerical data-AFRELI algorithm. *IEEE Trans. Fuzzy Syst.* 8(5), 591–600 (2000)
16. Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: Rawlin, G. (ed.) *Foundations of genetic Algorithms*, vol. 1, pp. 265–283. Morgan Kaufman, San Francisco (1991)
17. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. *Foundations of Genetic Algorithms 2*, 187–202 (1993)
18. Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. *IEEE Trans. Syst., Man, Cybern.* 22(6), 1414–1427 (1992)

Testing for Heteroskedasticity of the Residuals in Fuzzy Rule-Based Models

José Luis Aznarte M.¹ and José M. Benítez²

¹ Centre for Energy and Processes, MINES ParisTech, France

² Dept. of Computer Science and A.I., CITIC-UGR, University of Granada, Spain

Abstract. In this paper, we propose a new diagnostic checking tool for fuzzy rule-based modelling of time series. Through the study of the residuals in the Lagrange Multiplier testing framework we devise a hypothesis test which allows us to determine if the residual time series is homoscedastic or not, that is, if it has the same variance throughout time. This is another important step towards a statistically sound modelling strategy for fuzzy rule-based models.

1 Introduction

In general, once a model is built and estimated, it has to be evaluated. This is true in the Soft Computing framework as well as in the classical Statistics approach. By evaluating a model we understand to find out if the model satisfies a set of quality criteria that allow us to say if the interesting characteristics of the system under study are actually being captured by it or not.

Notwithstanding, this set of evaluation criteria is heavily dependent on several considerations: the final use that the model is built for, the inner characteristics of the system that are to be captured and whether the emphasis is put on the empirical behaviour of the model or if there are theoretical considerations that are considered to be more important. This is evident when we consider the evaluation means used in the Soft Computing field as opposed to those used in the statistical approach to time series analysis.

In the usually engineering-oriented Soft Computing framework, there has been an overwhelming preeminence of just one evaluation criterion, and this has been the *goodness of fit*. Generally, evaluation of a model consists on computing the prediction (or classification) error produced when it is faced with a previously unseen problem of the same type of the one used to estimate it. This measure, in its different flavours (mean squared error, mean average error and so on) is affected by some inherent limitations: it is not very meaningful for a single model unless compared against other models, and is usually range-dependent, which makes it difficult to compare the same model applied to different problems represented by data sets with different characteristics.

On the other hand, the evaluation in the statistical approach to time series has usually more to do with obtaining an estimate of the probability that the model is effectively capturing the interesting characteristics of the data set, and this is achieved through developing hypothesis tests, also known as misspecification tests.

There is a basic assumption behind modelling: a part of the system under study behaves according to a model but there is another part which cannot be explained by it and is usually considered to be white noise. This is the main idea encoded in the expression of the general model

$$y_t = G(\mathbf{x}_t; \boldsymbol{\psi}) + \varepsilon_t, \quad (1)$$

and it is also behind some of the diagnostic checking procedures.

It is interesting to obtain a precise knowledge about the series of the residuals, $\{\varepsilon_t\}$, by for example determining if its values are independent and normally distributed. If the residuals were not independent, that would mean that the model is failing to capture an important part of the behaviour of the series, and hence it should be respecified. This can be done through the test presented in [5].

Another desirable property that the model should satisfy refers to the variance of the series $\{\varepsilon_t\}$. If a model is properly capturing the inner behaviour of the series, the residuals should have the same variance at any point of the series. Failing to ensure this implies that the model's precision depends on time, and hence that there are parts of the state-space that are not properly modelled. This will affect very negatively to the performance of the model. Thus this situation should be properly detected so that convenient action for modelling is taken.

The current paper addresses the detection of this situation when fuzzy rule-based systems are used to model time series. The chosen procedure is through the definition of a hypothesis test, which we describe and do a preliminary evaluation.

2 Heteroskedasticity in Time Series Modeling

Ethymologically, heteroskedasticity means differing dispersion or variance. In statistics, a time series is called heteroskedastic if it has different variances throughout the time, and homoskedastic if it shows constant variance in the observable period.

Suppose we have a time series $\{y_t\}_{t=1}^n$ and a vector of time series (explanatory variables) $\{\mathbf{x}_t\}_{t=1}^n$. When considering conditional expectations of y_t given \mathbf{x}_t , the time series $\{y_t\}_{t=1}^n$ is said to be heteroscedastic if the conditional variance of y_t given \mathbf{x}_t changes with t . This is also referred as conditional heteroscedasticity to emphasize the fact that it is the series of conditional variance that changes and not the unconditional variance.

A graphical representation might help understand heteroskedasticity. The left part of figure [1], (which is adapted from [4]), depicts a classic picture of a homoskedastic situation. We can see a regression line estimated via orthogonal least squares in a simple, bivariate model. The vertical spread of the data around the predicted line appears to be fairly constant as X changes. In contrast, the right part of the figure shows a similar model with heteroskedasticity. The vertical spread of the errors is large for small values of X and then gets smaller as X rises. If the spread of the errors is not constant across the X values, heteroskedasticity is present.

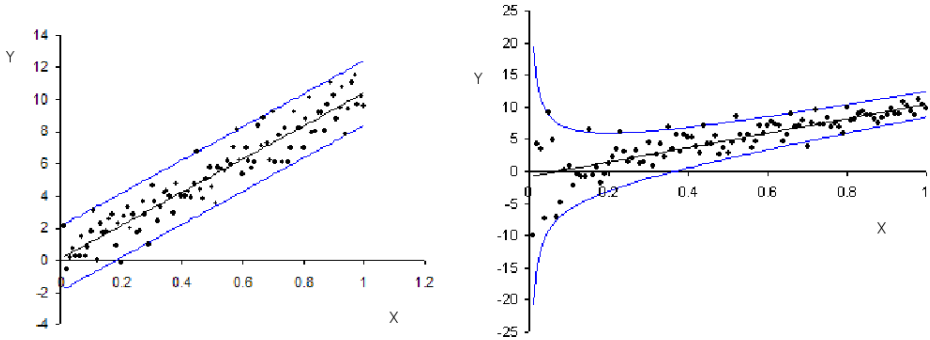


Fig. 1. Example of homoskedastic series (left) and heteroskedastic series (right)

In the case of fuzzy rule-based models for time series analysis, we might be interested in studying the heteroskedasticity of the residual series in the state-space regions defined by the antecedent of the rules. If our model’s residual series show smoothly changing variance between the rules, it is likely that some rules are failing to capture the behaviour of the series in their state-space subset. This represents an important source of diagnostic information about the goodness of the model.

3 Fuzzy Rule-Based Models for Time Series Analysis

When dealing with time series problems (and, in general, when dealing with any problem for which precision is more important than interpretability), the Takagi-Sugeno-Kang paradigm is preferred over other variants of FRBMs. When applied to model or forecast a univariate time series $\{y_t\}$, the rules of a TSK FRBM are expressed as:

$$\begin{aligned} &\text{IF } y_{t-1} \text{ IS } A_1 \text{ AND } y_{t-2} \text{ IS } A_2 \text{ AND } \dots \text{ AND } y_{t-p} \text{ IS } A_p \\ &\text{THEN } y_t = b_0 + b_1y_{t-1} + b_2y_{t-2} + \dots + b_p y_{t-p}. \end{aligned} \quad (2)$$

In this rule, all the variables y_{t-i} are lagged values of the time series, $\{y_t\}$.

Concerning the fuzzy reasoning mechanism for TSK rules, the *firing strength* of the i th rule is obtained as the t -norm (usually, multiplication operator) of the membership values of the premise part terms of the linguistic variables:

$$\omega_i(\mathbf{x}) = \prod_{j=1}^d \mu_{A_j^i}(x_j), \quad (3)$$

where the shape of the membership function of the linguistic terms $\mu_{A_j^i}$ can be chosen from a wide range of functions. One of the most common is the Gaussian bell, although it can also be a logistic function and even non-derivable functions as a triangular or trapezoidal function.

The overall output is computed as a weighted average or weighted sum of the rules output. In the case of the weighted sum, the output expression is:

$$y_t = G(\mathbf{x}_t; \boldsymbol{\psi}) + \varepsilon_t = \sum_{i=1}^R \omega_i(\mathbf{x}_t) \cdot \mathbf{b}_i \mathbf{x}_t + \varepsilon_t, \tag{4}$$

where G is the general nonlinear function with parameters $\boldsymbol{\psi}$, R denotes the number of fuzzy rules included in the system and ε_t is the series of the residuals as mentioned in the Introduction. While many TSK FRBMs perform a weighted average to compute the output, additive FRBMs are also a common choice. They have been used in a large number of applications, for example [8,9,10,16].

It has been proved [1] that this specification of the FRBM nests some models from the autoregressive regime switching family. More precisely, it is closely related with the Threshold Autoregressive model (TAR) [15], the Smooth Transition Autoregressive model (STAR) [14], the Linear Local-Global Neural Network (L²GNN) [13] and the Neuro-Coefficient STAR [12].

This relation has given place to an ongoing exchange of knowledge and methods from the statistical framework to the fuzzy rule-based modelling of time series. For instance, a linearity test against FRBM has been developed [6], and more contributions are yet to come.

In this paper we will consider two types of membership functions: sigmoid, μ_S , and Gaussian, μ_G . The sigmoid function is the one used in [12], and although it is not so common in the fuzzy literature, we will use it here as an immediate result derived from the equivalences stated in [2]. As we know, it is defined as

$$\mu_S(\mathbf{x}_t; \boldsymbol{\psi}) = \frac{1}{1 + \exp(-\gamma(\boldsymbol{\omega} \mathbf{x}_t - c))}, \tag{5}$$

where $\boldsymbol{\psi} = (\gamma, \boldsymbol{\omega}, c)$.

On the other hand, Gaussian function will also be used because it is the most common membership function in fuzzy models. It is usually expressed as

$$\mu_G(\mathbf{x}_t; \boldsymbol{\psi}) = \prod_i \exp\left(-\frac{(x_i - c_i)^2}{2\sigma^2}\right) \tag{6}$$

but we will rewrite it as

$$\mu_G(\mathbf{x}_t; \boldsymbol{\psi}) = \prod_i \exp(-\gamma(x_i - c_i)^2), \tag{7}$$

where $\boldsymbol{\psi} = (\gamma, \mathbf{c})$.

4 Test of Homoscedasticity of the Residuals of an FRBM

If an FRBM is properly identified and estimated, one might expect that the residuals have a normal distribution, $\varepsilon_t \sim N(0, \sigma^2)$. Moreover, it is expected that the residuals retain this distribution throughout time, that is, that the mean and the variance of ε_t remain constant through the changes of regime resulting from the prevalence of the different rules in different parts of the state-space.

It is hence interesting to develop a test which can determine if the variance σ^2 of the residual series changes when the model switches from one regime to

another or not. Assuming it does vary, we might note it as a time series σ_t^2 , whose specification would be:

$$\sigma_t^2 = \sigma^2 + \sum_{i=1}^r \sigma_i^2 \mu_{\sigma,i}(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_{\sigma,i}}) \tag{8}$$

where $\mu_{\sigma,i}$ are sigmoid or Gaussian function satisfying the identifiability restrictions defined in [4]. This formulation allows the variance to change smoothly between regimes.

Following [11], in order to avoid complicated restrictions over the parameters to guarantee a positive variance, we rewrite equation (8) as

$$\sigma_t^2 = \exp(G_\sigma(\mathbf{x}_t; \boldsymbol{\psi}_\sigma, \boldsymbol{\psi}_{\mu_{\sigma,i}})) = \exp\left(\varsigma + \sum_{i=1}^r \varsigma_i \mu_{\sigma,i}(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_{\sigma,i}})\right), \tag{9}$$

where $\boldsymbol{\psi}_\sigma = [\varsigma, \varsigma_1, \dots, \varsigma_r]'$ is a vector of real parameters.

To derive the test, let us consider $r = 1$. This is not a restrictive assumption because the test statistic remains unchanged if $r > 1$. We rewrite model (9) as

$$\sigma_t^2 = \exp(\varsigma + \varsigma_1 \mu_\sigma(\mathbf{x}_t; \boldsymbol{\psi}_{\mu_\sigma})), \tag{10}$$

where μ_σ is defined as (5) or as (7), depending on the membership function used by the model.

In both cases, sigmoid or Gaussian, the null hypothesis of homoscedasticity of the residuals is $H_0 : \gamma_\sigma = 0$. As usual, model (10) is only identified under the alternative $\gamma_\sigma \neq 0$ and we expand the membership function into a first-order Taylor expansion around $\gamma_\sigma = 0$. Replacing the function by its Taylor approximation and ignoring the remainder, both the sigmoid and the Gaussian case result in

$$\sigma_t^2 = \exp\left(\rho + \sum_{i=1}^q \rho_i x_{i,t}\right), \tag{11}$$

so the null hypothesis becomes $H_0 : \rho_1 = \rho_2 = \dots = \rho_q = 0$. Under H_0 , $\exp(\rho) = \sigma^2$.

The local approximation to the normal log-likelihood function in a neighbourhood of H_0 for observation t is

$$l_t = -\frac{1}{2} \ln(2\pi) - \frac{1}{2} \left(\rho + \sum_{i=1}^q \rho_i x_{i,t} \right) - \frac{\varepsilon_t^2}{2 \exp(\rho + \sum_{i=1}^q \rho_i x_{i,t})}. \tag{12}$$

In order to derive a LM type test, we need the partial derivatives of the log-likelihood:

$$\frac{\partial l_t}{\partial \rho} = -\frac{1}{2} + \frac{\varepsilon_t^2}{2 \exp(\rho + \sum_{i=1}^q \rho_i x_{i,t})}, \tag{13}$$

$$\frac{\partial l_t}{\partial \rho_i} = -\frac{x_i}{2} + \frac{\varepsilon_t^2 x_i}{2 \exp(\rho + \sum_{i=1}^q \rho_i x_{i,t})}, \tag{14}$$

and their consistent estimators under the null hypothesis:

$$\left. \frac{\partial \hat{l}_t}{\partial \rho} \right|_{H_0} = \frac{1}{2} \left(\frac{\varepsilon_t^2}{\hat{\sigma}^2} - 1 \right), \tag{15}$$

$$\left. \frac{\partial \hat{l}_t}{\partial \rho_i} \right|_{H_0} = \frac{x_{i,t}}{2} \left(\frac{\varepsilon_t^2}{\hat{\sigma}^2} - 1 \right), \tag{16}$$

where $\hat{\sigma}^2 = 1/T \sum_{t=1}^T \hat{\varepsilon}_t^2$. The LM statistic can then be written as

$$LM = \frac{1}{2} \left\{ \sum_{t=1}^T \left(\frac{\varepsilon_t^2}{\hat{\sigma}^2} - 1 \right) \tilde{\mathbf{x}}_t \right\}' \left\{ \sum_{t=1}^T \tilde{\mathbf{x}}_t \tilde{\mathbf{x}}_t' \right\}^{-1} \left\{ \sum_{t=1}^T \left(\frac{\varepsilon_t^2}{\hat{\sigma}^2} - 1 \right) \tilde{\mathbf{x}}_t \right\} \tag{17}$$

where $\tilde{\mathbf{x}}_t = [1, \mathbf{x}_t]'$. For details, see [11].

The test can be carried out in stages as follows:

1. Estimate model (4) assuming homoscedasticity and compute the residuals $\hat{\varepsilon}_t$. Orthogonalize the residuals by regressing them on $\nabla G(\mathbf{x}_t; \hat{\psi})$ and as before compute the $SSR_0 = \frac{1}{T} \sum_{t=1}^T \left(\frac{\hat{\varepsilon}_t^2}{\hat{\sigma}_{\hat{\varepsilon}_t}^2} - 1 \right)^2$, where $\hat{\sigma}^2$ is the unconditional variance of $\tilde{\varepsilon}_t$.
2. Regress $\left(\frac{\hat{\varepsilon}_t^2}{\hat{\sigma}_{\hat{\varepsilon}_t}^2} - 1 \right)$ on $\tilde{\mathbf{x}}_t$ and compute the residual sum of squares $SSR_1 = \frac{1}{T} \sum_{t=1}^T \tilde{\nu}_t^2$.
3. Compute the χ^2 statistic

$$LM_{\chi^2}^\sigma = T \frac{SSR_0 - SSR_1}{SSR_0}$$

or the F version of the test

$$LM_F^\sigma = \frac{(SSR_0 - SSR_1)}{s} \left(\frac{SSR_1}{(T - s - n)} \right)^{-1}.$$

Where T is the number of observations. Under H_0 , $LM_{\chi^2}^\sigma$ is asymptotically distributed as a χ^2 with s degrees of freedom and LM_F^σ has approximately an F distribution with s and $T - s - n$ degrees of freedom.

5 Empirical Evaluation

In this work we have performed a preliminary assessment of the properties of the test. In this line, we have considered three real-world time series, modeled them with FRMBs and then proceeded to their analysis.

The considered cases are fully described in [11], and are a well known ecology problem (the Lynx series), a planning/management problem and a botanic problem.

The first series, commonly referred to as the Lynx series, is composed of the annual records of lynx captures in a certain part of Canada during a period

Table 1. Results of misspecification tests for three models facing real world cases (significance value: 0.95)

| model | #rules | sigmoid membership function | | | Gaussian membership function | | |
|-------|--------|-----------------------------|--------|------------|------------------------------|--------|------------|
| | | σ_{ε_t} | AIC | p -value | σ_{ε_t} | AIC | p -value |
| A | 2 | 0.191 | -313 | 0.179 | 0.205 | -307 | 0.645 |
| B | 2 | 0.097 | -6590 | 0.000 | 0.098 | -6570 | 0.000 |
| C | 11 | 0.122 | -24357 | 0.234 | 0.120 | -24516 | 0.566 |

spanning 113 years. It is a common benchmarking series used to test and compare time series models, and here we have used its logarithmic transformation. An FRBM with two rules (model A) was identified following the iterative procedure proposed in [1], and it was later estimated using a Genetic Algorithm.

The second considered series comes from an emergency call center and is the record of the number of calls received daily throughout four years. As the series is non-stationary and shows a high variability, it was differenced after applying a log-transformation. The identified FRBM (model B) was also composed of just two fuzzy rules, which were also fine tuned through a Genetic Algorithm.

Finally, the third series was a daily aerobiological log obtained over sixteen years in the city of Granada (Spain), containing daily counts of airborne olive tree pollen grains. This series was previously studied in [3].

Table 1 shows some information about the application of the FRBM, both in its sigmoid and Gaussian versions, to the three time series mentioned above. More precisely, for each model, the table shows, the number of rules of the model, the values for the variance of the residuals (σ_{ε_t}) and the Akaike information criterion (AIC), together with the p -value obtained with the test for homoscedasticity of the residuals.

By studying the p -values shown in columns 5 and 8 we can see how the null hypothesis of the test was rejected in all the six cases, which leads us to conclude that the variance of the residuals remained constant through time in every application.

As mentioned above, this is a necessary condition for considering that a model is properly capturing the behaviour of a time series.

6 Conclusions and Final Remarks

In this paper, a new statistical tool to evaluate the residuals of a fuzzy rule-based model has been presented. It consists of a test against homoskedasticity of the residuals, that is, a test that allow the user to determine if the variance of the residual series remains constant through time. In other words, this test is able to tell if a model’s errors are bigger in some parts of the state-space or not.

This represents a useful contribution and another step towards a statistically sound framework for the use of fuzzy rule-based models.

Acknowledgements. This work has been partially funded by Spanish Ministerio de Ciencia e Innovación (MICINN) under Project grants MICINN TIN2009-14575 and CIT-460000-2009-46.

References

1. Aznarte M., J.L.: Modelling time series through fuzzy rule-based models: a statistical approach. Ph.D. thesis, Universidad de Granada (2008)
2. Aznarte M., J.L., Benítez, J.M., Castro, J.L.: Smooth transition autoregressive models and fuzzy rule-based systems: Functional equivalence and consequences. *Fuzzy Sets Syst.* 158(24), 2734–2745 (2007)
3. Aznarte M., J.L., Benítez, J.M., Nieto-Lugilde, D., de Linares Fernández, C., de la Díaz Guardia, C., Alba Sánchez, F.: Forecasting airborne pollen concentration time series with neural and neuro-fuzzy models. *Expert Systems with Applications* 32(4), 1218–1225 (2007)
4. Aznarte M., J.L., Benítez Sánchez, J.M.: On the identifiability of TSK additive fuzzy rule-based models. In: *Soft Methods for Integrated Uncertainty Modelling. Advances in Soft Computing*, vol. 6, pp. 79–86. Springer, Heidelberg (2006)
5. Aznarte M., J.L., Benítez Sánchez, J.M.: Testing for linear independence of the residuals in the framework of fuzzy rule-based models. In: *Ninth International Conference on Intelligent Systems Design and Applications (ISDA)*, Pisa (Italy) (December 2009)
6. Aznarte M., J.L., Medeiros, M., Benítez Sánchez, J.M.: Linearity testing against a fuzzy rule-based model. *Fuzzy Sets and Systems*, doi:dx.doi.org/10.1016/j.fss.2010.01.005
7. Barreto, H., Howland, F.: *Introductory Econometrics: Using Monte Carlo Simulation with Microsoft Excel*. Cambridge University Press, Cambridge (2005)
8. Byun, H., Lee, K.: A decision support system for the selection of a rapid prototyping process using the modified topsis method. *Intern. Journal of Advanced Manufacturing Technology* 26(11-12), 1338–1347 (2005)
9. John, R., Innocent, P.: Modeling uncertainty in clinical diagnosis using fuzzy logic. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 35(6), 1340–1350 (2005)
10. Lee, I., Kosko, B., Anderson, W.F.: Modeling gunshot bruises in soft body armor with an adaptive fuzzy system. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 35(6), 1374–1390 (2005)
11. Medeiros, M., Veiga, A.: Diagnostic checking in a flexible nonlinear time series model. *Journal of Time Series Analysis* 24, 461–482 (2003)
12. Medeiros, M., Veiga, A.: A flexible coefficient smooth transition time series model. *IEEE Transactions on Neural Networks* 16(1), 97–113 (2005)
13. Suarez-Farinas, M., Pedreira, C.E., Medeiros, M.C.: Local global neural networks: A new approach for nonlinear time series modeling. *Journal of the American Statistical Association* 99, 1092–1107 (2004), <http://ideas.repec.org/a/bes/jnlasa/v99y2004p1092-1107.html>
14. Teräsvirta, T.: Specification, estimation and evaluation of smooth transition autoregressive models. *J. Am. Stat. Assoc.* 89, 208–218 (1994)
15. Tong, H.: On a threshold model. *Pattern Recognition and Signal Processing* (1978)
16. Vieira, C.F., Palma, L.B., da Silva, R.N.: Robust fault diagnosis approach using analytical and knowledge based techniques applied to a water tank system. *International journal of engineering intelligent systems for electrical engineering and communications* 13(4), 237–244 (2005)

Heuristic Methods Applied to the Optimization School Bus Transportation Routes: A Real Case

Luzia Vidal de Souza and Paulo Henrique Siqueira

Department of Graphic Expression, Federal University of Parana, PO Box 19081, Curitiba, Parana 81.531-970, Brazil
luzia@ufpr.br, paulohs@ufpr.br

Abstract. The problem discussed in this paper is similar to the Vehicle Routing Problem (VRP), however new contributions are proposed. In this work a heuristic algorithm is proposed to determine the set of the Bus Stops. A new approach is proposed to construct digital maps containing the roads where the vehicles will be able to travel, since there are no digital maps of these regions. The real distances between the points are calculated and the heuristics Location Based Heuristic with some additional features was used to propose the new routes. The algorithm was named by Adapted Location Based Heuristic (ALBH). The School Transportation Problem was implemented in the State of Parana for 399 cities. We present here the results obtained for 10 of the 399 cities. The results obtained by using this approach showed improvement in daily distance performed and in the amount of the vehicles used to do the job.

Keywords: Vehicle Routing Problem, School Bus Transportation, Heuristics Methods.

1 Introduction

Applications of Operations Research in the transportation field always produce great impact, because these techniques are able to improve the quality of pick-up and delivery service and they can reduce the operating costs of the systems considered. An important application is the School Transportation Problem, which is a special case of vehicle routing with time windows (VRPTW), heterogeneous fleet and considering simultaneous pick up and delivery. In these kinds of problems a set of vehicles make the pick up and / or delivery products or people to consumers dispersed in an area. The goal is to find a set of vehicle routes and schedules that satisfies a variety of constraints and minimizes the total fleet operating costs [1], [2].

The development of models that offer optimal solutions to this problem is complex, because it is a problem that considers many constraints and the computational cost to do this task is high, sometimes it becomes impossible to be performed. For this reason many efforts have been made by researchers in worldwide to find new approaches that can produce good solutions to such problems with low computational cost [3]. Constrains considering vehicle capacity, maximum distance of each route, time windows and minimum coverage of the breakpoints.

In this paper we applied some techniques of Operations Research and heuristic algorithms to solve the real problem of the School Transportation in the Brazilian state of Parana. Using this methodology the manager of the Bus School Transport will be able to improve the service by reducing the time of the students inside the vehicles while minimize the total distance travelled by all the vehicles. In this problem the stop points are mixed, in other words students from different schools and degrees can be picked up and dropped by the same route, since the objective of minimizing the total distance of the route and attend a larger number of students can be hold.

The VRP appears in many applications such as garbage collect, distributions of drinks, gasoline and other products. The School Transportation Problem can be faced like a VRP considering a heterogeneous fleet, consists of determining a set of routes and schedule each vehicle to a route respecting the vehicle capacity constrains. The demand points are the student homes, and constrains to determine the routes are the vehicles capacity, the maximum time spent for each student into the vehicle may be limited by maximum length of the route. This paper was carried out in three steps. The first step is to determine the breakpoints (Bus Stops - BS) of the vehicles, considering the maximum distance that students can walk from their homes until these points. In the second stage the distances between two Bus Stop and/or School Points (SP) are calculated. Finally, in the third step we applied the Location Based Heuristic (LBH) with some proposed adaptations to be used in the real situation, and it was called Adapted Location Based Heuristic (ALBH) to route and schedule the buses so as minimizing total operating costs respecting the constrains to find a feasible solution for the problem.

2 Literature Review

In the literature few records of practical applications of theories of VRP to the School Bus Transportation are found. In [4] students are assigned to an intersection of streets adjacent to streets from their homes, and a subset of these points is considered potential points to solve the traditional routing problem.

In [5] Tabu Search is used to solve the traditional routing problem, showing results by solving several problems of the literature and presenting comparisons of their results with other techniques, but in general the solved problems are small.

A formulation of Integer Linear Programming to solve the School Bus Transportation is presented in [6], with appropriate restrictions in a Flemish region, where some comparisons and partial solutions to small problems are shown by testing feasible solutions and comparing the computational time by using this technique and similar others.

Some techniques that can be used to solve the VRP is shown in [7], including traditional techniques such as Clark Wright savings, 2-stage methods, and even metaheuristics like Tabu Search. Results are presented comparing all the techniques shown in the article, using as parameters both the computational time as the quality of each solution.

Reference [8] show 3 techniques to solve the problem of School Bus Transportation in New Jersey City, they are: the Clarke and Wright Savings, the computer program

called Router and a Sweep method. The results of these three techniques show that is possible to find a good solution for the city used in the tests.

In [9] a self-organizing Neural Network is used to solve both the problem of Multiple Salesman as the problem of vehicle routing with capacity constraints. The Neural Network is compared with similar techniques in the literature and the results are promising.

In their paper Braca et al [3] present a methodology to solve the School Bus Transportation and it was applied to New York City. The algorithm to solve this problem was proposed by Bramel and Simchi-Levi [10], [11] and converges asymptotically to the optimal solution to vehicle routing problem capacity. In this problem the authors used 838 breakpoints and 73 schools. The minimum number of vehicles determined by the algorithm was 59 to be used in the morning and 56 in the afternoon. In this work we choose the Braca's algorithm because the similarity between the problems, and this method was applied to solve the School Bus Transportation 399 cities of Parana, but we present here the results only for 20 of them, because the difficulty to compare the results. Some modifications were implemented to adapt this technique to the real problem.

3 The School Bus Problem

A set of students dispersed in an area must be picked up at your bus stop and dropped at your school every school day. After the bus stops have been determined the students must be assigned to the bus stop nearest from their homes the next step is to route and schedule the vehicles minimizing the total distances daily travelled and then reducing the time that the students spend inside the vehicle while the safety requirements are satisfied.

The real case that we considered here is about a Brazilian State of Parana, covering 399 cities, the customers are students in elementary and high school. They must be picked up in their bus stop and dropped off at their schools. The school bus transportation costs are calculated considering the total distance daily travelled and the amount and type of vehicles used to do the job. Usually, the fleet is heterogeneous, because there are regions where some vehicles are unable to traffic according to the road conditions, which generally are very steep and narrow, making impossible that certain kind of vehicles travel trough these roads. In this Province the School Bus Transportation is under responsibility of the Municipal Departments of Education, who is responsible to contract the vehicles that will be used to take the students at their schools.

The goal is to assign the students to the break points, find the better route and schedule the vehicle that will be used in this route, minimizing total distance traveled and the amount of used vehicles. The mathematical formulation for this problem is presented below, in accordance to the equations (1) to (12):

$$\text{Minimize } \sum_{i \in V} \sum_{j \in V} \sum_{k=1}^k c_{ij} x_{ijk} \quad (1)$$

$$\text{Subject to: } \sum_{j \in V_E} \sum_{k=1}^k y_{jk} \leq K \tag{2}$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik}, \forall i \in V, k = 1, 2, \dots, K \tag{3}$$

$$\sum_{l \in S} w_{hl} = \sum_{i \in V \setminus V_E} y_{hk}, h \in V_E, k = 1, 2, \dots, K \tag{4}$$

$$\sum_{k=1}^K y_{ik} \leq 1, \forall i \in V \setminus V_E \tag{5}$$

$$\sum_{i \in V} \sum_{l \in S} z_{ilk} \leq C_k, k = 1, 2, \dots, K \tag{6}$$

$$z_{ilk} \leq y_{ik}, \forall i, l, k \tag{7}$$

$$\sum_{i \in V} \sum_{k=1}^K z_{ilk} = 1, \forall l \in S \tag{8}$$

$$y_{ik} \in \{0,1\}, \forall i \in V, k = 1, 2, \dots, K \tag{9}$$

$$x_{ijk} \in \{0,1\}, \forall i, j \in V, i \neq j \tag{10}$$

$$z_{ilk} \in \{0,1\}, \forall i, j \in V, i \neq j \tag{11}$$

$$w_{hl} \in \{0,1\}, \forall h \in V_E, \forall l \in S \tag{12}$$

where, c_{ij} = distance between i and j ; K = amount of vehicles; C_k = capacity of the vehicle k ; V = set of customers to be visited (bus stop); V_E = set of the depots (school points); S = set of customers (set of students); $x_{ijk} = 1$ if arc(i, j) belongs to the route operated by vehicle k and 0 otherwise; $y_{ik} = 1$ if the client i is visited by vehicle k and 0 otherwise; $z_{ilk} = 1$ if the customer l is picked-up by the vehicle k at the breakpoint i and 0 otherwise; $w_{hl} = 1$ if the client l goes to school h and 0 otherwise.

The objective function (1) seeks to minimize the total distance traveled by all the vehicles. Constrains (2) guarantee that all the vehicles started its route at the depot (school). Constrains (3) ensure that if the customer i is visited by the vehicle k , so an arc must be traveled by the vehicle k by starting and departure from the same point i . Constrains (4) guarantee that all the customers will be in the route of this respective schools. Constrains (5) ensure that each client is visited by exactly one vehicle except the schools. Constrains (6) define that the capacity of the vehicles will not be exceeded. Constrains (7) guarantee that the customer l is not collected at the point i by the vehicle k if the vehicle k does not pass through to the point i . Constrains (8) guarantee that each customer will be collected only once. Constrains from (9) to (12) define the decision variables.

4 The School Bus Problem

To solve this problem we considered three steps that are described below:

Step 1: Determination of the buses stops - In this stage the stop points are defined in accordance with the geographical position of the student's home. Each student should be assigned to their nearest bus stop. In this stage some parameters must be considered and they are defined by Municipal Departments of Education of the State and they are:

- d_{max} : Maximum distance that the student must walk from this home until the bus stop - this parameter is limited by a minimum and maximum value it must be between 500 and 3,000 meters;
- Distance between the buses stops: the distance between the buses stop must be between 1,000 and 3,000 meters;
- *Cover*: Minimum distance between the customers (students) to their depot (school). This parameter should be between 1,000 and 3,000 meters, that means if customer lives less than this distance from the depot he should not use the school bus transportation.

Fig. 1 displays the buses stops proposed by the algorithm and the geographic positions of the customers and the depots in Castro city for the morning period, using $d_{max} = 1,500$ meters.

Before building the routes, is necessary to calculate the real distances between the buses stops and the depots (schools). To calculate these distances it was necessary to build the roads, because there are no digital maps available for these cities. The dataset were provided by the State Ministry of Urban Development in Parana containing the geographical coordinate points of the pathways of 100 to 100 meters for each city and the information from crosses and end of the roads. Using these datasets it was possible to generate the road map that shows the available ways that might be explored in searching better solutions for problem. The datasets of each city is extremely large, which makes the process slow, for example the database for Castro City contains 43,793 points beside the bus stops and depots points. The pseudo code of the algorithm to determine the buses stop is presented below.

Algorithm to find out the buses stops

```
{Let:  $U$  set of the customers that were not assigned;  $P(.)$  the buses stops index;  $d_{max}$  maximum distance between customers and their buses stops;  $dist(i, j)$  distance from the customer (student home)  $i$  until the bus stop  $j$ ; cover: minimum distance between the customer until his school};
  Select  $U^*$  customers  $i$  such as  $dist(i, school(i)) \leq cover$ ; Let
   $U \leftarrow U \setminus U^*$ ;
  {Choose randomly the geographic position of each customer  $i \in U$ . This position is the bus stop  $j$ ;
   Select customers  $k$  such as  $dist(k, j) \leq d_{max}, \forall k P(k) = j$ .
  }
```

Step 2: Calculating the Real distances between the buses stops - Although the computational cost to calculate the real distances are often greater than the cost to calculate the Euclidean distances it is really necessary in real cases. Let A_i the set of the adjacent points to the point i and T_k the set of the other points in the same road k . The pseudo code to calculate the shortest distance between two points is presented below.

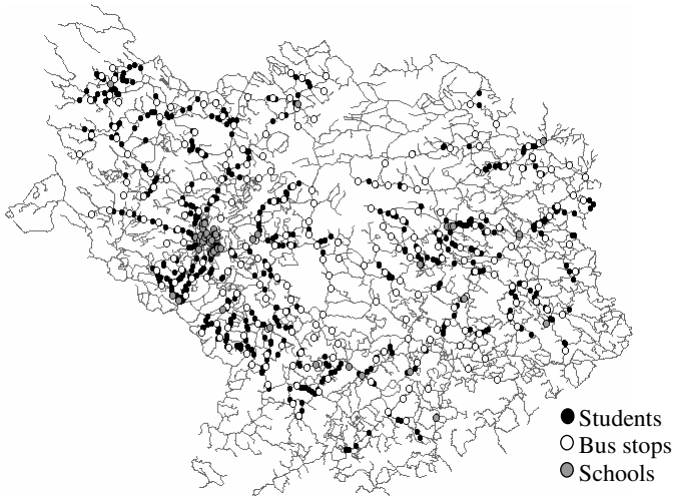


Fig. 1. Map of Castro City – Bus Stops (morning)

Calculate of the real distances between two points

```
{Let:  $i$  the initial point,  $j$  the final point;  $dist(.)$  function to
calculate real distance between two points;  $R$  set of sequence of
points between  $i$  and  $j$  into the route;  $n = 1$ };
While  $n \leq 2$  select the set of the adjacent points from the point  $i$ ,  $A_i$ 
and calculate  $dist(m, j) \forall m \in A_i$ ;
  {Let  $k$  such as  $dist(k, j) = \min_{m \in A_i, m \in R} \{dist(m, j)\}$ ,  $R \leftarrow R \cup \{k\}$ 
  {Select one of 4 possible alternatives to the point  $k$ :
  - If the point  $k$  is the end of road, find  $R^* = \{l, \dots, k\}$ , such as
 $R^* \subset R$  and  $l$  is the initial point of the cross-road between  $k$ 
and  $l$ .  $R \leftarrow R \setminus R^*$ , and  $i = l$ . In next iterations, points of the  $R^*$ 
must be prohibits for the route;
  - If point  $k$  make a cycle, find  $R^* = \{k, \dots, k\}$ , such as  $R^* \subset R$ 
determine the part of the route that contains the cycle. Let
 $R \leftarrow R \setminus R^*$  and  $i = k$ . In next iterations the points of set  $R^*$  must
be prohibits in the route;
  - If  $k$  is not a crossroad, find  $T_k = \{k_1, k_2, \dots, k, \dots, k_t\}$ , where  $k_1$ 
and  $k_t$  are crossroads. Let  $R \leftarrow R \cup T_k$ , and  $i = k_t$ 
  - If  $k$  is crossroad, let  $i = k$ 
  If  $k = j$ ,  $R$  is the full route from the point  $i$  to  $j$ ;
  If  $n = 1$ , let  $i \leftarrow j$  and  $j \leftarrow i$ ,  $R^{**} \leftarrow R$ ,  $R = \emptyset$  and  $n \leftarrow n + 1$ ;
  }
}
```

Choose between R and R^{**} the shortest route.

Step 3: Building the routes - ALBH – Adapted Location Based Heuristic - After the assignment of the customers (students) has been made to their bus stops and the real distances have been calculated is time to construct the routes. The used technique in this work was based in Location Based Heuristic (LBH) and it was called Adapted Location Based Heuristic (ALBH), which converges asymptotically to the optimal solution in Vehicle Routing Problems Capabilities and heterogeneous fleets. The pseudo code to calculate the shortest distance between two points is presented below.

Design of the routes - ALBH

```

Let v the biggest vehicle that was not assigned to a route;
{Let m=0 and S={1,2,...,l} the set of the all customers in their
bus stops that are not into a route. Select m∈S, such as
dist(m,school(m))= maxi∈S{dist(i,school(i))}. While S≠∅:
{Select the index of the farthest student from his school that
is not in the route j∈S, such as dist(j,school(j))=
maxi∈S{dist(i,school(i))}. Let S←S\{j}. Calculate the longer
route allowed for the student j, where:
n1=  $\frac{\text{number of the students}}{\text{capability of the vehicles}}$ ,
n2=  $\frac{\text{dist}(m, \text{school}(m)) - \text{dist}(j, \text{school}(j))}{\text{dist}(m, \text{school}(m))}$ 
longer_route=dist(j,school((j))(1+(1+n1)n2)
Let the part of the route Rm={j→school(j)}. Repeat while ck=∞:
{For each student i∈S, calculate
ci=comp_route(Rm,i,school(i)).
Let ck=mini∈S{ci}. If ck < ∞ then:
Let Rm←route(Rm,k,school((k)). Let S←S\{k};
}
}
If the number of customers in Rm is less or equal than Cv and
comp_route(Rm)≤ mlonger_route:
m = m+1;
Otherwise
The heuristic solution of vehicle v is {R1,R2,...,Rm};
Check if the used vehicle is completely full, if there are many
places not used, select another vehicle that was not routed yet and
that is more appropriate to the route.
}

```

The function *comp_route(.)* is used to calculate the total distance by inserting the point *i* and their respective school in the route *R*. The position of the inserted point is exchanged until the shortest path in the route has been found. The function *route(.)* insert the customer *k* and his depot (school) in the best position defined by the function *comp_route(.)*. The most important modifications proposed in ALBH algorithm are: vehicles have different capacities *C_v*; the routes start with the farthest point from its school; the vehicle assignment for this route is the biggest vehicle and after the construction of the route is checked if there is a smallest vehicle available that can be assigned for the route. The *n₁* term is used to allow longer routes when the number of customers in the buses is less than the total demand of the route. The *n₂* term is used to avoid very long routes and to prohibit that the maximum length of the route is limited by the distance between the farthest bus stop and its school. If this term is not used many routes can be created containing only one stop point because its forbidden to pick another students in this same route. The methodology is used separately for each period (morning, afternoon and night) and the daily total distance for each vehicle is calculated considering both taking the students from their homes to school and from the school to their homes.

5 Experiments and Results

The experiments were accomplished using the data base from ten cities in the state of Parana, they are. The results were compared to the real situation. The real distances to make the comparisons were calculated using obtained by using a GPS system (Global Positioning System) for mapping the School Bus routes that are actually performed in each city.

Table 1 shows the dataset for each city about the amount of students using bus school transportation, vehicles and schools. The amount of available vehicles in each city is showed in the 3rd column of the Table 1. Fig. 2 shows the design of the routes on the Castro maps for the morning period. In this simulation the longer distance used between the stop points is 1,500 meters and the *cover* (distance between the customer and depot) is 2,000 meters. The used technique was ALBH which provided an economy ranging between 8.8 and 58.8% in the total daily distance traveled as shown in Table 1.

The School Transportation Manager can set the parameters that better adapt to the reality of that city. The choice of these parameters have great influence in the optimization process, for example if the longer route is less than 20 km there will be a greater demand on the amount of vehicles while longer routes can raise the time of the students inside the vehicle.

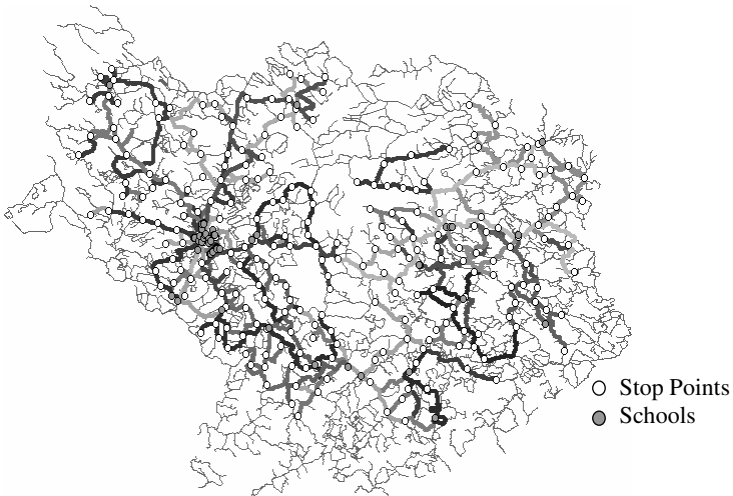


Fig. 2. Map of Castro city – routes in the morning

5.1 Results

To evaluate the performance of our approach, we compared the obtained results with the real situation it means that the total mileage traveled by the fleet was compared to the mileage that is being held in each city by the manager of the school bus transportation. The results were not compared to another algorithm because this problem had not been solved before using any technique. In the first column of table 1 are listed the name of the cities for which the simulation was performed. In the other columns of the

Table 1 are showed the dataset about each city and the results are in the 8th column. The computational time (last column) depends on the how large is the city, the number of the students and schools. The main advantages of using this methodology are: decrease the time that the student spend into the bus by reducing the amount of the bus stops; decrease in the costs by reducing the number of the vehicles used and the total daily distance traveled in each period as is showed in the Table 1. The economy ranges between 1% and 49%. In this experiment the best solution was obtained when the biggest distance between the customers to his depot is of 2,000 meters.

Table 1. Results obtained using ALBH for 10 cities

| City | schools | vehicles | range | students | bus stop | km | economy(%) | time(sec) |
|------------------|---------|----------|-------|----------|----------|-----------|------------|-----------|
| Apucarana | 89 | 52 | GPS | 1712 | 1128 | 2,613,291 | - | - |
| | | | 500m | 1643 | 246 | 2,366,212 | 9 | 3322 |
| | | | 2000m | 1174 | 211 | 2,063,966 | 21 | 1057 |
| Araucária | 33 | 80 | GPS | 2440 | 1393 | 3,388,274 | - | - |
| | | | 500m | 2379 | 379 | 3,367,108 | 1 | 7718 |
| | | | 2000m | 1864 | 314 | 2,954,370 | 13 | 2321 |
| Bom Jesus do Sul | 8 | 11 | GPS | 471 | 218 | 537,872 | - | - |
| | | | 500m | 471 | 57 | 500,066 | 7 | 133 |
| | | | 2000m | 397 | 49 | 456,570 | 15 | 77 |
| Castro | 49 | 52 | GPS | 4501 | 2711 | 7,553,586 | - | - |
| | | | 500m | 4409 | 678 | 6,889,292 | 9 | 33884 |
| | | | 2000m | 3824 | 640 | 6,718,128 | 11 | 8541 |
| Lapa | 44 | 23 | GPS | 2160 | 652 | 2,675,360 | - | - |
| | | | 500m | 2078 | 200 | 2,523,292 | 6 | 17952 |
| | | | 2000m | 1759 | 187 | 2,303,492 | 14 | 7872 |
| Londrina | 160 | 107 | GPS | 2291 | 1387 | 5,285,160 | - | - |
| | | | 500m | 2169 | 391 | 4,880,536 | 8 | 10522 |
| | | | 2000m | 1594 | 337 | 4,790,098 | 9 | 3082 |
| Maringá | 103 | 23 | GPS | 652 | 395 | 1,792,030 | - | - |
| | | | 500m | 615 | 128 | 1,111,740 | 38 | 723 |
| | | | 2000m | 372 | 99 | 916,102 | 49 | 353 |
| Medianeira | 32 | 31 | GPS | 1316 | 936 | 1,366,637 | - | - |
| | | | 500m | 1278 | 143 | 1,292,976 | 5 | 1282 |
| | | | 2000m | 878 | 123 | 1,198,736 | 12 | 322 |
| Ponta Grossa | 165 | 59 | GPS | 1407 | 659 | 3,431,116 | - | - |
| | | | 500m | 1343 | 252 | 3,136,768 | 9 | 6202 |
| | | | 2000m | 1126 | 235 | 3,054,114 | 11 | 3262 |
| Prudentópolis | 72 | 66 | GPS | 1818 | 1090 | 3,359,002 | - | - |
| | | | 500m | 1766 | 319 | 3,150,630 | 6 | 9577 |
| | | | 2000m | 1419 | 270 | 2,887,784 | 14 | 5864 |

6 Conclusions and Future Works

In this paper we proposed a new methodology for the School Bus Transportation and it was performed to ten cities of Brazilian state of Parana. The problem was solved in three phases, the first one is about to determine the better positions of the Buses Stop; the second is calculate the real distances between all the points and the last one is use the ALBH to build the routes that can better solve the problem.

The tests were performed using a Pentium IV computer, 2.8 GHz, 1Gb of ram memory. The computational time is not high considering the amount of the points in

each data set. The computational time is calculated considering the all process for each period (morning, afternoon and night) in each city. This all system must be performed at most twice a year, in the beginning and in the middle of the year in order to consider changes the address students. During this time any insertion or exclusion of the bus stop should be made manually to avoid huge changes in the routes. The requirements to perform this process are: dataset of the cities containing the geographic positions of the students, the schools and the number of the available vehicles to be used, this allows that the optimization process can be easily applied to anywhere by making few adaptations. It is important to note that no digital map is required. To perform the process is only necessary to have the coordinates of the points and the proposed methodology is able to construct the roads. In the future work we intend to compare the results obtained by using the ALBH algorithm and the proposed methodology with another approach like Neural Network.

References

1. Bodin, L., Golden, B., Assad, A., Ball, M.: The state of the art in the routing and scheduling of vehicles and crews. *Computers and Operations Research* 10, 63–211 (1983)
2. Fisher, M.L.: Vehicle routing. In: *Handbooks in Operations Research and Management Science: Network Routing*, vol. 8, pp. 1–33 (1992)
3. Braca, J., Bramel, J., Posner, B., Simchi-Levi, D.: A computerized approach to the New York City school bus routing problem. *IIE Transactions* 29, 693–702 (1997)
4. Dulac, G., Ferland, J., Fogues, P.A.: School bus routes generator in urban surroundings. *Computers and Operations Research* 7, 199–213 (1980)
5. Toth, P., Vigo, D.: The Granular Tabu Search and Its Application to the Vehicle-Routing Problem. *INFORMS Journal on Computing* 15, 333–346 (2003)
6. Schittekat, P., Sevaux, M., Sorensen, K.: A mathematical formulation for a school bus routing problem. In: *ICSSSM 2006 - International Conference on Service Systems and Service Management*, pp. 1552–1557. IEEE Press, New York (2006)
7. Laporte, G., Gendreau, M., Potvin, J.Y., Semet, F.: Classical and modern heuristics for the vehicle routing problem. *Internat. Transact. Internat. Transact. in Operational Research* 7, 285–300 (2000)
8. Spasovic, L., Chien, S.: A Methodology for Evaluating of School Bus Routing - A Case Study of Riverdale. In: *80th Annual Meeting of Transportation Research Board*, New Jersey, pp. 1–17. Transportation Research Board, Washington (2001)
9. Modares, A., Somhom, S., Enkawa, T.A.: A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems. *Journal of International Transactions in Operational Research* 6, 591–606 (1999)
10. Bramel, J., Simchi-Levi, D.: A location based heuristic for general routing problems. *Operations Research* 43, 649–660 (1995)
11. Simchi-Levi, D., Bramel, J.: On the optimal solution value of the capacitated vehicle routing problem with unsplit demands. Working Paper. Columbia University (1990)

Particle Swarm Optimization in Exploratory Data Analysis

Ying Wu¹ and Colin Fyfe²

¹ Coastal and Marine Resources Centre,
University College Cork, Ireland
y.wu@ucc.ie

² Applied Computational Intelligence Research Unit,
The University of West of Scotland, Scotland
colin.fyfe@uws.ac.uk

Abstract. We discuss extensions of particle swarm based optimization (PSO) algorithms in the context of exploratory data analysis. In particular, we apply these extensions to principal component analysis, exploratory projection pursuit and topology preserving mappings. Our extensions include combining PSO algorithms with stochastic sampling and a form of reinforcement learning known as Q-learning. We illustrate on a variety of artificial data sets and show that our new results are better than previous results on such data sets.

1 Introduction

Particle Swarm Optimization (PSO) [11,12], based on the Swarm Intelligence algorithm [2], has been widely studied recently as a stochastic search technique for global optimization over continuous problem spaces. The PSO algorithm was motivated as simulating the behaviors of a population of simple agents (ants, birds or even people) interacting locally with each other and with their environment. Even when there is no central control of how an individual simple agent should behave, the local interactions between these agents can lead to a globally optimized behavior. During recent years, PSO algorithms have encompassed a broad variety of problems such as artificial intelligence [4], clustering [6], Markov Decision Processes [5], etc.

Exploratory data analysis is a set of methods with which we try to extract as much information as possible from a data set of high dimension and huge volume. In this paper, we will develop PSO-based algorithms for exploratory data analysis. We first derive a PSO-based method for projection problems, such as principal component analysis (PCA) and exploratory projection pursuit (EPP). The results show that our algorithm can identify the optimal solutions quickly with high accuracy, even when the size of data set is small, which often leads to poor generalization, and the number of iterations is low. Then we incorporate the PSO algorithm with Q-learning, a form of reinforcement learning [14], and apply the PSO-based Q-learning algorithm to solve the PCA problem. We compare the results with those we report in [17,1]. We demonstrate that the PSO-based

algorithm has better performance in that not only is the accuracy of the final results improved, but also the number of iterations required to achieve the global optimum are reduced. Finally, we illustrate a topology preserving mapping with the PSO algorithm.

We illustrate our new methods on simple processes such as PCA and EPP but this is for didactic purposes only and is a continuation of our previous investigations into non-standard optimizations [17,11]. We envisage that the results of these investigations will feed into more complex optimizations in future.

2 Particle Swarm Optimization

Particle Swarm algorithms were developed on the basis of the astonishing self-organization exhibited by groups of very simple agents which may be insects, birds or animals. Such agents appear to exhibit group behavior which far transcends the individual thought processes of the agents and the algorithms attempt to emulate such emergent behaviors. Particle Swarm Optimization (PSO) [11,12] is a stochastic search algorithm, which aims to identify the global optimum of an objective function without requiring any gradient rule.

In PSO, a swarm of simple agents (particles) is initialized in the multidimensional problem space with random positions X_i and velocities V_i . These particles are considered to move in the problem space searching for the global optimal solution and the location of each particle in the problem space represents one possible solution. Thus when a particle moves to another location, a different solution to the problem is generated. In each iteration, the solutions represented by the particles are evaluated by a fitness function f , then the locations $X_i = (\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{im})$ and velocities $V_i = (\mathbf{v}_{i1}, \mathbf{v}_{i2}, \dots, \mathbf{v}_{im})$ of these particles are adjusted using

$$V_i(t+1) = \omega \cdot V_i(t) + C_1 \cdot \varphi_1 \cdot (P_{i1} - X_i(t)) + C_2 \cdot \varphi_2 \cdot (P_g - X_i(t)) \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

where the variables φ_1 and φ_2 are random positive numbers, drawn from a uniform distribution, C_1 and C_2 are called *acceleration constants* and ω is called the *inertia weight*. The variable P_{i1} is the local best solution found so far by the i^{th} particle and the variable P_g is the location of the particle with highest fitness during the previous iterations, called the global best solution. We can see that the movement of particles encompasses two impulses. The first is called the cognitive behavior where one particle follows its own cognitive experience via its own optimal personal local solution foregoing the group solution. The second is to consider the social behavior in which each particle gets attracted to the group's center. Therefore, at the end of the simulation, most of the particles will converge to a small ball surrounding the global optimum of the search space.

3 PSO in Exploratory Data Analysis

Exploratory data analysis is a set of methods with which we try to extract as much information as possible often from a data set of high dimension and

huge volume. In this section, we apply the PSO algorithm to a set of projection methods. We consider that each particle is deemed to be taking movements in an environment that consists of data to be explored, in order to maximize the fitness in this environment. We use stochastic units, \mathbf{W} , drawn from a Gaussian distribution to sample the population of particles which means each unit represents one possible solution and the samples/units \mathbf{W} are all drawn from $\mathcal{N}(\mathbf{m}, \beta^2 I)$, the Gaussian distribution with mean \mathbf{m} and variance β^2 . It is these parameters, the mean \mathbf{m} and variance β^2 that the particle swarm optimization algorithm produces and this algorithm is directed by the efficiency of the individual samples. Although different projection methods have their own objective functions, they share the common property that the fitness function determines how well one particle fits its environment. We thus show our method is quite a general one that can be easily applied to different projection methods.

3.1 Principal Component Analysis

Principal Component Analysis (PCA) finds the linear filters W onto which projections of a data set have greatest variance. Thus each particle represents one possible filter and is evaluated in proportion to its variance. At iteration t , we create an initial M particles, $\mathbf{w}_1^t, \dots, \mathbf{w}_M^t$ from $\mathcal{N}(\mathbf{m}, \beta^2 I)$, a D -dimensional isotropic Gaussian distribution with centre, \mathbf{m} . We calculate the fitness value for each of these particles and the particle with highest fitness value is identified as the local best solution \mathbf{w}^{t*} . At each iteration, the whole swarm keeps a memory of the best particle \mathbf{w}^* visited so far by all the particles, which is known as the global best solution. We move the distribution of the stochastic unit with the learning rule

$$\mathbf{m} \leftarrow (1 - \eta)\mathbf{m} + \eta(C_1 \cdot \varphi_1 \cdot (\mathbf{w}^{t*} - \mathbf{m}) + C_2 \cdot \varphi_2 \cdot (\mathbf{w}^* - \mathbf{m})) \tag{3}$$

$$\beta^2 \leftarrow (1 - \eta)\beta^2 + \eta \left(\frac{\sqrt{\sum_{i=1}^t (\mathbf{w}^{i*} - \hat{\mathbf{w}}^{T*})^2}}{t} \right) \tag{4}$$

where η is the learning rate. The two variables, C_1 and C_2 are acceleration constants, which are used to effect the stochastic nature of the algorithm and scaled by constants $0 < C_1, C_2 < 2$. The value of $C_1\varphi_1$ controls the degree of local interactions in the current population of particles and the value of $C_2\varphi_2$ controls the degree of global interaction. The variable $\hat{\mathbf{w}}^{T*}, T = 1, 2, \dots, t$ is the average value of all the local best solutions we have found during the past t iterations.

We summarize our PSO-based PCA algorithm as follows:

1. Select one item of data from the data set randomly.
2. Generate a population of particles $\mathbf{w}_1^t, \dots, \mathbf{w}_M^t$ from the currently estimated distribution $\mathcal{N}(\mathbf{m}_t, \beta_t^2)$.
3. Evaluate each particle according to the fitness function and identify the local best solution \mathbf{w}^{t*} .

4. Compare the fitness values of the local best solution, \mathbf{w}^{t*} and the current global best solution and identify the new global solution, \mathbf{w}^* .
5. Update the parameters of the distribution to make a new probability density function with (3) and (4).
6. If less than maximum number of iterations, go back to step 1.

To illustrate our algorithm, we create a 5-dimensional artificial data set of 10000 samples, whose elements are drawn independently from Gaussian distributions with $x_i \sim \mathcal{N}(0, \sigma_i^2)$, so x_5 has the greatest variance and x_1 has the lowest variance. The fitness function is defined as $f = \frac{1}{1 + \exp(-\gamma|\mathbf{w}^T \mathbf{x}|)}$. To identify multiple components, we use the Gram-Schmidt method as the deflation method. Thus, from the second component onwards, right after generating a particle \mathbf{w}_j^t from the currently estimated distribution $\mathcal{N}(\mathbf{m}_t, \beta_t^2 I)$, we subtract $(\mathbf{w}_j^T \mathbf{m}_k) \mathbf{m}_k, k = 1, 2, \dots, j - 1$ from \mathbf{w}_j . We set $\varphi_1 = 1$ and $\varphi_2 = 1.2$ and the learning rate $\eta = 0.01$, which is reduced linearly to zero. The number of iteration is 10000.

Table 1. The weights from the artificial data experiment for five principal components by the PSO-based PCA method

| | | | | | |
|-----|----------------|----------------|---------------|---------------|----------------|
| PC1 | -0.0004 | -0.0004 | 0.0000 | 0.0010 | -1.0000 |
| PC2 | -0.0001 | 0.0000 | -0.0008 | 1.0000 | 0.0010 |
| PC3 | 0.0013 | -0.0009 | 1.0000 | 0.0008 | 0.0000 |
| PC4 | 0.0023 | -1.0000 | -0.0009 | 0.0000 | 0.0004 |
| PC5 | -1.0000 | -0.0023 | 0.0013 | -0.0001 | 0.0004 |

We see in Table 1 that the five principal components have been identified with very high accuracy and our algorithm converges smoothly and quickly as shown in Figure 1.

3.2 Exploratory Projection Pursuit

Exploratory Projection Pursuit (EPP) (see [748]) defines a group of techniques designed to investigate structure in high dimensional data sets by finding “interesting” directions in the data space.

Table 2. The weights from the artificial data experiment for EPP by the PSO-based EPP method

| | | | | | |
|-----|---------|--------|---------|---------|---------------|
| EC1 | -0.0033 | 0.0157 | -0.0012 | -0.0070 | 0.9998 |
|-----|---------|--------|---------|---------|---------------|

To illustrate our method for exploratory projection pursuit, we create 1000 samples of 5 dimensional data in which 4 elements of each vector are drawn from $\mathcal{N}(0, 1)$, while the fifth contains data with negative kurtosis: we draw this

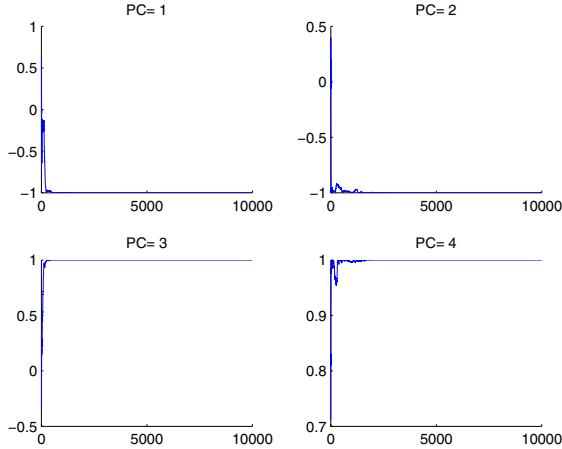


Fig. 1. Convergence of the PCA weight vectors to the optimal directions by the PSO-based PCA method. The vertical axis shows the cosine of the angle between the current filter and the optimal filter. The horizontal axis shows the number of iterations.

also from $\mathcal{N}(0, 1)$, but randomly add or subtract 5. Before performing the algorithm, we sphere the data set so that it has zero mean and unit variance in all directions. We use $S(\mathbf{w}) = |\tanh(\mathbf{w}^T \mathbf{x})|$ as the fitness function. Note that $g(s) = |\tanh(s)|$ is an even function that can be used to measure kurtosis. Table 2 shows the outcome of the simulation. We can see that the distribution with negative kurtosis has been identified with high accuracy and extremely quickly by the defined fitness function. Convergence was as fast and stable as for the PCA experiment.

3.3 Principal Component Analysis with Q-Learning

Reinforcement learning [14] is a sub-area of machine learning which trains agents by reward and punishment without needing to specify how to achieve a task. Reinforcement learning algorithms attempt to find a policy that maps states to actions so as to maximize some notion of long-term reward.

In reinforcement learning models, an agent exists within its environment and at each time of interaction, t , the agent perceives the state $s_t \in S$ of the environment and the set of possible actions $A(s_t)$. Then the agent chooses an action $a \in A(s_t)$ that changes the state of the environment from s_t to s_{t+1} and receives a reward r_{t+1} . The agent's behavior, B , should be based on a policy π , mapping states to actions, that tends to maximise the long-term sum of values of the rewards.

The Q-learning method has been introduced in [15, 16]. This method directly approximates the optimal action-value function, Q^* , by the learned action-value function, Q , and the best possible action selected in the subsequent state:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)].$$

In [17], we derived a method to solve PCA problems with Q-learning. In this section, we incorporate the PSO algorithm with Q-learning and apply the PSO-based Q-learning algorithm to solve PCA problem.

The state of the system at any time is the data sample presented to the system at that time, i.e. we equate s_t with \mathbf{x}_t , $s_t = \mathbf{x}_t$. We use a parametric (actually Gaussian) estimator for the statistic to be calculated and the action taken in response to the state is to sample the weight vector \mathbf{w} from the distribution $\mathcal{N}(\mathbf{m}, \beta^2 I)$ with the current estimate of the parameters, \mathbf{m} and β^2 , having been optimized in the previous iteration. Then we may identify rewards and update both the estimated Q-value (which is what we wish to maximise) of the estimator and the parameters (mean and variance) of the estimator.

We use the same 5 dimensional data as before in which the first principal component is readily identified as the fifth input dimension. At each iteration, the particles, \mathbf{W} , are drawn from the Gaussian distribution, $\mathcal{N}(\mathbf{m}, \beta^2 I)$, which are used to represent the possible actions given the current state/data point. For the PCA problem, the reward of each possible action is defined by $r = \frac{1}{1 + \exp(-\gamma|\mathbf{w}^T \mathbf{x}|)}$. Then the algorithm identifies the locally best particle, \mathbf{w}^{t*} , with the highest reward r^{t*} . Meanwhile, the algorithm also keeps a memory of the globally best particle, \mathbf{w}^* , with the highest Q-value Q^* so far. The local best particle \mathbf{w}^{t*} with its reward r^{t*} is used to calculate the new Q-value of the state/data point by the following

$$\Delta Q_i^{t*} \leftarrow \alpha(r^{t*} + \gamma Q^* - Q_i) \tag{5}$$

$$Q_i \leftarrow Q_i + \Delta Q_i^{t*} \tag{6}$$

In each iteration, the calculation is then followed by

$$\mathbf{m} = (1 - \eta)\mathbf{m} + \eta \cdot \Delta Q_i \cdot (C_1 \cdot \varphi_1 \cdot (\mathbf{w}^{t*} - \mathbf{m}) + C_2 \cdot \varphi_2 \cdot (\mathbf{w}^* - \mathbf{m})) \tag{7}$$

where i is the index of current data point we have randomly selected, D is the dimensionality of the data and η is the learning rate.

The number of iterations is 5000 and the learning rate is initialized to 0.01 and reduced linearly to zero. Again the algorithm identifies the optimal principal component direction very quickly with high accuracy. We compare the performance of this algorithm with the results by immediate reward reinforcement learning and the Q-learning method that we show in [17] in Figure 2. It is clear that with the same number of iterations, the PSO-based PCA algorithm converges to the optimal solution but the other two methods fail to do so.

3.4 Topology Preserving Manifolds

A topographic mapping captures some structure in the data set, so that points which are mapped close to one another have some common feature while points

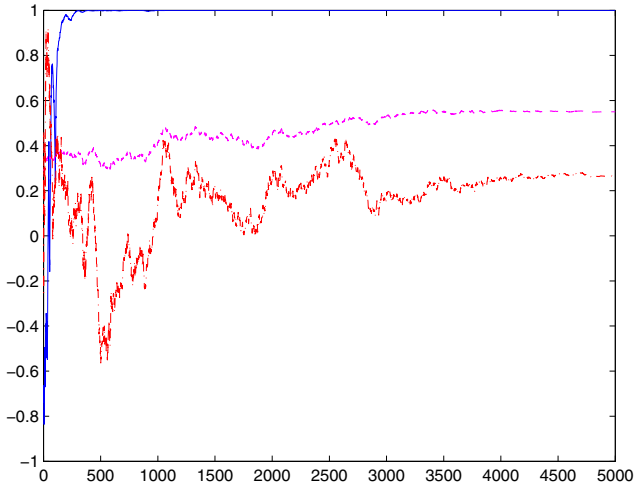


Fig. 2. Convergence of the PCA weight vector to the optimal directions. Solid line: by PSO-based Q-learning algorithm in this section. Magenta Dashed line: by immediate reinforcement learning method. Red dot dashed line: by standard Q-learning method.

that are mapped far from one another do not share this feature. The most common topographic mapping is Kohonen's self-organizing map (SOM) [13]. The Generative Topographic Mapping (GTM) [3] is a mixture of experts model which treats the data as having been generated by a set of latent points where the mapping is *non-linear*. In [9], we have derived an alternative topology preserving model, called the *Topographic Products of Experts* (ToPoE), based on products of experts [10], which is closely related to the generative topographic mapping. In this section, we present a PSO-based method to perform the topology preserving mapping.

Given a set of data points $\mathbf{t}_1, \dots, \mathbf{t}_N$, we follow [3,9] to create a latent space of points $\mathbf{x}_1, \dots, \mathbf{x}_K$ which lie equidistantly on a line or at the corners of a grid. To allow non-linear modeling, we define a set of M basis functions, $\phi_1(), \dots, \phi_M()$, with centres μ_j in latent space. Thus we have a matrix Φ where $\phi_{kj} = \phi_j(\mathbf{x}_k)$, each row of which is the response of the basis functions to one latent point, or, alternatively each column of which is the response of one of the basis functions to the set of latent points. Typically, the basis function is a squared exponential. These latent points are then mapped to a set of points $\mathbf{m}_1, \dots, \mathbf{m}_K$ in data space where $\mathbf{m}_j = (\Phi_j \mathbf{W})^T$, through a set of weights, \mathbf{W} . The matrix \mathbf{W} is $M \times D$ and is the sole parameter which we change during training. We have

$$\mathbf{m}_k = \sum_{j=1}^M \mathbf{w}_j \phi_j(\mathbf{x}_k) = \sum_{j=1}^M \mathbf{w}_j \exp(-\beta \|\mu_j - \mathbf{x}_k\|^2), \forall k \in \{1, \dots, K\}. \quad (8)$$

where $\phi_j(), j = 1, \dots, M$ are the M basis functions, and \mathbf{w}_j is the weight from the j^{th} basis function to the data space. The algorithm is summarized as:

1. Randomly select a data point, \mathbf{t}_n .
2. Find the closest prototype, say \mathbf{m}_{k^*} , to \mathbf{t}_n .
3. Generate T particles from the Gaussian distribution, $\mathcal{N}(\mathbf{m}_{k^*}, \beta_{k^*}^2 I)$. Call the particles, $\mathbf{y}_{k^*,1}, \dots, \mathbf{y}_{k^*,T}$. We note that we are using $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K$ to perform two conceptually separate functions, as prototypes or means to which the data will be quantized and as centres of Gaussian distributions from which samples will be drawn.
4. Evaluate the particles using $S(\mathbf{y}) = \exp(-\gamma \|\mathbf{y} - \mathbf{t}_n\|^2)$ as the fitness function.
5. Identify the local best particle, \mathbf{y}_{k^*,t^*} , which has the largest fitness value.
6. Update the parameters

$$\mathbf{w} \leftarrow \mathbf{w} + \eta [C_1 \cdot \varphi_1 \cdot (\mathbf{y}_{k^*,t^*} - \mathbf{m}_{k^*})\phi(\mathbf{x}_{k^*}) + C_2 \cdot \varphi_2 \cdot (\mathbf{w}^* - \mathbf{w})] \quad (9)$$

where η is the learning rates and \mathbf{y}_{k^*,t^*} is the local best particle and \mathbf{w}^* is the global best solution we have found up to the current iteration.

7. Update the prototypes' positions using [\(8\)](#).
8. Evaluate the total distance between prototypes and data points and identify the global best solution, \mathbf{w}^* , which is defined in [\(10\)](#).

$$\mathbf{w}^* \in \left\{ \arg \min_{\mathbf{w} \in W} \sum_{k=1}^K \sum_{n=1}^N \|\mathbf{m}_k - \mathbf{t}_n\|^2 \right\} \quad (10)$$

It is worth noting that for topology preserving mappings, the global optimal solution we look for is *a set of weights* through which the latent points in the latent space are projected to prototypes in the data space and the distance between these prototypes and all the data points is minimized. In our algorithm above, given one data point \mathbf{t}_n selected at one iteration, the particles are drawn around the prototype \mathbf{m}_{k^*} that is closest to the selected data point \mathbf{t}_n , and the local best particle \mathbf{y}_{k^*,t^*} in step [5](#) represents a new location of the prototype \mathbf{m}_{k^*} that is possibly closer to the data point \mathbf{t}_n . Since the particles do not represent the weights, the particles do not represent a topology preserving mapping, but the local best particle can be regarded as a *local best solution*. We thus do not keep a memory of the global best particle as the standard PSO does. Instead we keep a memory of the global best solution \mathbf{w}^* in step [8](#), which is defined in formula [\(10\)](#) and in step [6](#), and we update the weights by considering the local best solution \mathbf{y}_{k^*,t^*} and the global best solution \mathbf{w}^* .

Figure [3](#) shows the result of a simulation in which there are 20 latent points lying equally spaced in a one dimensional latent space, passed through 5 basis functions and mapped to the data space by the linear mapping W . We generate 1000 2-dimensional data points, (x_1, x_2) , from the function $x_2 = x_1 + 1.25 \sin(x_1) + \rho$, where ρ is the noise from a uniform distribution in $[0, 1]$. Hence the data set, though 2 dimensional, has an implicit dimensionality of 1. The number of iterations is 10000. The latent points' projections are shown in the figure as black *s and are adjacent latent points' projections are joined with a line. We clearly see that the one dimensional nature of the data has been identified and that neighbouring latent points have responsibility for neighbouring data points.

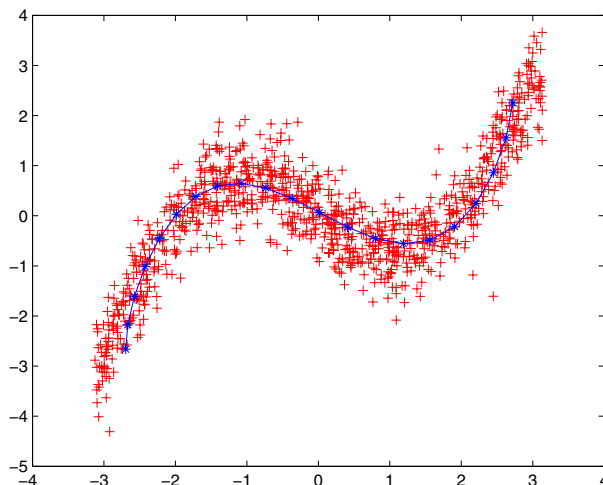


Fig. 3. The data are shown by ‘+’s and the latent points’ projections are ‘*’s

4 Conclusion

In this paper, we have illustrated the use of PSO-based algorithm for exploratory data analysis. For the linear projection problems, we define the population of particles at one iteration by a set of parameters, \mathbf{W} sampled by stochastic units drawn from $\mathcal{N}(\mathbf{m}, \beta^2 \mathbf{I})$, the Gaussian distribution with mean \mathbf{m} and variance β^2 . We demonstrated the PSO-based algorithm on principal component analysis and exploratory projection pursuit. The results have shown that the PSO-based algorithms can identify the optimal direction quickly, robustly and with high accuracy. Then we incorporate the PSO-based algorithm into the Q-learning method, where the new algorithm is used to solve the PCA problem. We demonstrated the new algorithm can converge to the global optimum more quickly with higher accuracy compared with the results we have in [17,1] by the immediate reward reinforcement learning and the Q-learning method. Finally, we have developed a PSO-based algorithm for topology preserving mappings. Instead of considering the local best particle and global best particle in the standard PSO algorithm, we use the local best solution and global best solution to update the weights through which the latent points in the latent space are projected to prototypes in the data space.

References

1. Barbakh, W., Wu, Y., Fyfe, C.: Non-standard parameter adaptation for exploratory data analysis. Springer, Berlin (2009)
2. Beni, G., Wang, U.: Swarm intelligence in cellular robotic systems. In: NATO Advanced Workshop on Robots and Biological Systems, Il Ciocco, Tuscany, Italy (1989)

3. Bishop, C.M., Svensen, M., Williams, C.K.I.: GTM: The generative topographic mapping. *Neural Computation* 10, 215–234 (1998)
4. Carvalho, M., Ludermir, T.B.: Particle swarm optimization of feed-forward neural networks with weight decay. In: *Proceedings of the Sixth International Conference on Hybrid Intelligent Systems*, p. 5 (2006)
5. Chang, H.S.: An adaptation of particle swarm optimization for markov decision processes. In: *IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1643–1648 (2004)
6. Cui, X., Potok, T.E., Palathingal, P.: Document clustering using particle swarm optimization. In: *Proceeding of Swarm Intelligence Symposium*, pp. 185–191 (2005)
7. Friedman, J.H.: Exploratory projection pursuit. *Journal of the American Statistical Association* 82(397), 249–266 (1987)
8. Fyfe, C.: *Hebbian Learning and Negative Feedback Networks*. Springer, Heidelberg (2005)
9. Fyfe, C.: Two topographic maps for data visualization. *Data Mining and Knowledge Discovery* 14, 207–224 (2007)
10. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Technical Report 2000-004, Gatsby Computational Neuroscience Unit, University College, London (2000)
11. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of IEEE International conference on Neural Networks*, vol. 4, pp. 1942–1948 (1995)
12. Kennedy, J., Eberhart, R., Shi, Y.: *Swarm Intelligence*. Morgan Kaufmann Academic Press, San Francisco (2001)
13. Kohonen, T.: *Self-organising maps*. Springer, Heidelberg (1995)
14. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge (1998)
15. Watkins, C.J.C.H.: *Learning from Delayed Rewards*. PhD thesis, Cambridge University (1989)
16. Watkins, C.J.C.H., Dayan, P.: Q-learning. *Machine Learning* (1992)
17. Wu, Y.: Non-standard adaptation of linear projections for exploratory data analysis. PhD thesis, University of The West of Scotland (2008)

Using the Bees Algorithm to Assign Terminals to Concentrators

Eugénia Moreira Bernardino¹, Anabela Moreira Bernardino¹,
Juan Manuel Sánchez-Pérez², Juan Antonio Gómez-Pulido²,
and Miguel Angel Vega-Rodríguez²

¹ Research Center for Informatics and Communications, Department of Computer Science, School of Technology and Management, Polytechnic Institute of Leiria, 2411 Leiria, Portugal
{eugenia.bernardino, anabela.bernardino}@ipleiria.pt

² Department of Technologies of Computers and Communications, Polytechnic School, University of Extremadura, 10071 Cáceres, Spain
{sanperez, jangomez, mavega}@unex.es

Abstract. With the recent growth of communication networks, a large variety of combinatorial optimization problems appeared. One of these problems is the Terminal Assignment Problem. The main objective is to assign a given set of terminals to a given set of concentrators. In this paper, we propose the Bees Algorithm to assign terminals to concentrators. The algorithm performs a kind of neighbourhood search and uses a local search method to locate the global minimum. The Bees Algorithm is a swam-based optimization algorithm that mimics the natural behaviour of honey bees. We show that the Bees Algorithm is able to achieve feasible solutions to Terminal Assignment instances, improving the results obtained by previous approaches.

Keywords: Communication Networks, Optimization Algorithms, Bees Algorithm, Terminal Assignment Problem.

1 Introduction

In recent years we have witnessed a tremendous growth of communication networks, which resulted in a large variety of combinatorial optimization problems. One of these problems is the Terminal Assignment (TA) Problem. In centralized computer networks, a central computer services several terminals or workstations. In a large network, some concentrators are used to increase the network efficiency. A collection of terminals is connected to a concentrator and each concentrator is connected to the central computer. The TA problem involves determining which terminals will be serviced by each concentrator. The number of concentrators and terminals and their locations are known. Each concentrator is limited in the amount of traffic that it can accommodate. For that reason, each terminal must be assigned to one node of the set of concentrators, in a way that no concentrator oversteps its capacity [1][2][3]. In this work, the problem is interpreted as a multi-objective task. The optimization goals are to simultaneously produce feasible solutions, to minimize the distances between concentrators and terminals assigned to them and to maintain a balanced distribution of

terminals among concentrators. The TA problem is a NP-complete combinatorial optimization problem. It means that the TA problem cannot be solved to optimality within polynomially bounded computation times.

This paper presents an application of a population-based optimization algorithm called the Bees Algorithm. The Bees Algorithm is inspired by the food foraging behaviour of honey bees [4] and uses a neighbourhood search method and a local search method to be able to locate the global minimum. The Bees Algorithm has been successfully applied to different optimization problems [5] including the training of neural networks for control chart pattern recognition, finding multiple feasible solutions to preliminary design problems, identification of wood defects, overcoming the local optimum problem of the K-means clustering algorithm, job scheduling, manufacturing cell formation, multi-objective optimization, optimizing the design of mechanical components, tuning a fuzzy logic controller for a robot gymnast, computer vision, image analysis and others.

Our algorithm is based on the Bees Algorithm proposed by Pham et al. in [4]. Embedded in the Bees Algorithm we use a Local Search (LS) algorithm proposed by Bernardino et al. [6], which is used to improve the quality of the solutions.

We compare the performance of the Bees Algorithm with four algorithms: Tabu Search (TS) Algorithm, Local Search Genetic Algorithm (LSGA), Hybrid Differential Evolution (HDE) Algorithm and Hybrid Ant Colony Optimization (HACO) Algorithm, used in literature.

This paper is structured as follows. In Section 2 we present the definition of the TA problem; in Section 3 we describe the Bees Algorithm implemented; in Section 4 we discuss the computational results obtained and, finally, in Section 5 we report the conclusions.

2 Terminal Assignment Problem

The TA problem involves the determination of which terminals will be serviced by each concentrator [1]. In the TA problem a communication network will connect N terminals and each with T_i demand via M concentrators and each with C_j capacity. No terminal's demand exceeds the capacity of any concentrator. A terminal site has a fixed and known location $CT_i(x,y)$. A concentrator site has also a fixed and known location $CP_j(x,y)$.

In this work, the solutions are represented using integer vectors. We use the terminal-based representation (see Fig. 1). Each position in the vector corresponds to a terminal. The value carried by the position i of the vector specifies the concentrator to which the terminal i is to be assigned.



Fig. 1. Terminal Based Representation

Fig. 2 illustrates an assignment to a problem with $N = 10$ terminal sites and $M = 3$ concentrator sites. The figure shows the coordinates for the concentrators, terminal sites and also their capacities.

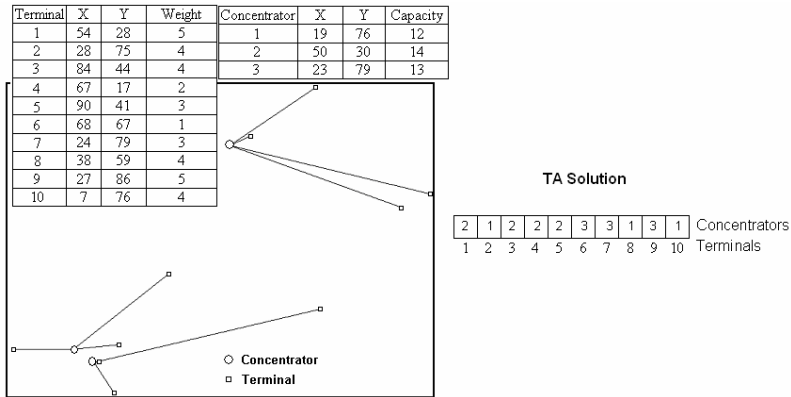


Fig. 2. TA Problem – example

3 Bees Algorithm

Swarm Intelligence (SI) is an Artificial Intelligence technique involving the study of collective behaviour in decentralized systems. Four widely known approaches are Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and Bees Algorithm. All these approaches can be used in real-world optimization problems. The Bees Algorithm is a new population based algorithm that mimics the natural behaviour of a swarm of bees [4]. The ABC algorithm also simulates the behaviour of bees, but uses a different algorithm model.

A SI algorithm is a population based algorithm. It starts with a population of individuals (i.e. potential solutions). These individuals are then manipulated over many iteration steps by mimicking the social behaviour of insects or animals, in an effort to find the optimal solution in the space of the problem solution. A potential solution “flies” through the search space by modifying itself according to its past experience and its relationship with other individuals in the population and the environment [7].

In Bees Algorithm [5], after creating the initial population of bees (ns), bees are ranked according to the fitnesses of their sites. The best nss of the ns sites are classified as “selected sites” and the best nbs of the nss sites are classified as “best sites”. The nbs bees are sent to the “best sites” and the nbb bees are sent to the remaining $(nss - nbs)$ “selected sites”. These bees (nbs and nbb) produce new sites in the neighbourhood of the “selected sites”. Searches in the neighbourhood of the best sites, which represent more promising solutions, are made by recruiting more bees to follow them than the other selected bees. The remaining bees ($ns - nss$) are classified as scout and assigned randomly.

Our algorithm is based on the Bees Algorithm proposed by Pham et al. in [4]. The basic form of the algorithm uses a neighbourhood search to explore around the selected sites. To improve the performance of the algorithm, we incorporate a LS algorithm proposed by Bernardino et al. [6]. The LS is used to improve the quality of the solutions in the population.

The main steps of the Bees Algorithm are given below:

```

Initialize Parameters
Create initial Population of bees,  $P^0$ 
Evaluate Population  $P^0$ 
WHILE stop criterion isn't reached
  Apply Local Search Procedure to all individuals in  $P^t$ 
  Select nss best bees in  $P^t$ ,  $PB^t$  t=iteration
  Compute probabilities for nsb best sites
  Compute probabilities for (nss-nbs) selected sites
  FOR i=1 to nss DO
    IF i<=nbs THEN
      Compute number of bees (nb) recruited for best site i
    ELSE
      Compute number of bees (nb) recruited for selected site i
  FOR j=1 to nb DO
    beej = NeighbourhoodSearch ( $PB^t_i$ )
    Evaluate beej
    Select best bee:
      IF fitness (beej) < fitness( $PB^t_i$ ) THEN
         $P^t_i$  = beej
      ELSE
         $P^t_i$  =  $PB^t_i$ 
  FOR i=nss+1 to ns DO //Assign remaining bees
    Create solution (bee),  $P^t_i$ 
    Evaluate  $P^t_i$ 

```

Initialization of parameters

The following parameters must be defined by the user (1) mi – number of iterations; (2) ns - number of initial bees; (3) nss – number of sites selected out of ns visited sites; (4) nbs – number of best sites out of nss selected sites; (5) $nbbs$ – number of bees recruited for the best nbs sites; (6) $nbss$ – number of bees recruited for the remaining selected sites ($nss-nbs$) and (7) nm - number of modifications.

Initial Population

The initial population (P^0) can be created randomly or in a deterministic form. The deterministic form is based in the Greedy Algorithm proposed by Abuali et al. [8]. The Greedy Algorithm assigns terminals to the closest feasible concentrator.

Evaluation of solutions

To evaluate how good a potential solution is relative to other potential solutions we use a fitness function. The fitness function returns a number (fitness value) that reflects how optimal the solution is. The fitness function is based on: (1) the total number of terminals connected to each concentrator (the purpose is to guarantee a balanced distribution of terminals among the concentrators); (2) the distance between the concentrators and the terminals assigned to them (the purpose is to minimize the distances between the concentrators and respective assigned terminals); (3) the penalization if a solution is not feasible (the purpose is to penalize the solutions when the total capacity of one or more concentrators is overloaded). The purpose is to minimize the fitness function.

The fitness function is based on the fitness function used in [2]:

$$fitness = 0,9 * \sum_{c=1}^M bal_c + \quad (1) \quad c(t) = \text{concentrator of terminal } t$$

$$0,1 * \sum_{t=1}^N dist_{t,c(t)} + \quad (2) \quad t = \text{terminal}$$

Penalization

$$bal_c = \begin{cases} 10 & \text{if } (total_c = \text{round}(\frac{N}{M}) + 1) \\ 20 * \text{abs}(\text{round}(\frac{N}{M}) + 1 - total_c) & \text{otherwise} \end{cases} \quad (3) \quad total_c = \sum_{t=1}^N \int_0^1 \text{if}(c(t)=c)$$

$$dist_{t,c(t)} = \sqrt{(CP[c(t)].x - CT[t].x)^2 + (CP[c(t)].y - CT[t].y)^2} \quad Penalization = \begin{cases} 0 & \text{if (Feasible)} \\ 500 & \text{otherwise} \end{cases}$$

Local Search

At the beginning of each iteration, our algorithm applies the LS procedure to the solutions in the population. The LS algorithm consists on applying a partial neighbourhood examination. We generate a neighbour by swapping two terminals between two concentrators $c1$ and $c2$ (randomly chosen). The algorithm searches for a better solution in the initial set of neighbours. If the better neighbour improves the actual solution, then the LS algorithm replaces the actual solution with the better neighbour. Otherwise, the algorithm creates another set of neighbours. In this case, one neighbour results in assigning one terminal of $c1$ to $c2$ or $c2$ to $c1$. The neighbourhood size is $N(c1) * N(c2)$ or $N(c1) * N(c2) + N(c1) + N(c2)$.

The LS algorithm consists on the following steps:

```

c1 = random (number of concentrators)
c2 = random (number of concentrators)
NN = neighbours of ACTUAL-SOL (one neighbour results of
interchange one terminal of c1 or c2 with
one terminal of c2 or c1)

SOLUTION = FindBest (NN)
IF Fitness(ACTUAL-SOL) < Fitness(SOLUTION) THEN
  NN = neighbours of ACTUAL-SOL (one neighbour results of
assign one terminal of c1 to c2 or c2 to c1)
  SOLUTION = FindBest (NN)
  IF Fitness(SOLUTION) < Fitness(ACTUAL-SOL) THEN
    ACTUAL-SOL = SOLUTION
ELSE
  ACTUAL-SOL = SOLUTION

```

The evaluation process is the step that consumes more time, which is usually the case in many real-life problems. Our LS procedure has some important improvements compared to the LS proposed by Bernardino et al. [6]. After creating a neighbour, the algorithm does not perform a full examination to calculate the new fitness value; it only updates the fitness value based on the modifications that were made to create the neighbour.

Select best bees

The bees that have the smallest fitnesses are chosen as “selected bees” (PB^t) and the sites visited by them are selected for neighbourhood search.

Compute probabilities for nbs best sites and

A bee is recruited for a best site i , depending on the probability value associated with that site. The probabilities are calculated by the following expression:

$$totalFitness = \sum_{n=1}^{nbs} fitness(PB_n^t)$$

$$p_i = \frac{totalFitness - fitness(PB_i^t)}{totalFitness}$$

Compute probabilities for $(nss - nbs)$ selected sites

A bee is recruited for a selected site i , depending on the probability value associated with that site. The probabilities are calculated by the following expression:

$$totalFitness = \sum_{n=nbs+1}^{nss} fitness(PB_n^t)$$

$$p_i = \frac{totalFitness - fitness(PB_i^t)}{totalFitness}$$

Compute number of bees

In our implementation the algorithm computes the number of bees, which will be sent to a site, according to previously determined probabilities:

nb_i = number of bees sent to site i .

IF $i \leq nbs$ *THEN*

$nb_i = p_i * nbbs$

ELSE

$nb_i = p_i * nbss$

Neighbourhood Search

The algorithm conducts searches in the neighbourhood of the selected sites, assigning more bees to search near to the best nbs sites.

The general mechanism of the neighbourhood search is represented in the next pseudo-code:

```

FOR n=1 TO nm DO
  t = random(N)
  closestC=1
  FOR c=1 TO M DO /*find the closest concentrator*/
    IF distance (t, c) < distance (t, closestC)
      closestC=c
  IF capacityFree (closestC) >= L(t)
    and mantainBalanced(closestC) THEN
    Assign terminal t to concentrator closestC
  ELSE
    cond=true
    REPEAT
      t1 = random(N)
      t2 = random(N)
      c1 = solution (t1)
      c2 = solution (t2)
      IF ( capacityFree(c2) - L(t2) >= L(t1) and
          capacityFree(c1) - L(t1) >= L(t2) ) and

```

```

        ( distance(t2,c1) <= distance(t1,c1) or
          distance(t1,c2) <= distance(t2,c2) ) THEN
      Assign t1 to c2 and t2 to c1
      cond = false
    WHILE cond=true

```

A neighbour is obtained by performing multiple moves which length is specified as nm (number of modifications). The algorithm performs nm modifications to find a new solution. First the algorithm chooses a random terminal t and searches the closest concentrator. If the concentrator has enough capacity and maintains a balanced distribution of terminals then the terminal t is assigned to the closest concentrator, $closestC$. Otherwise, the algorithm generates two random terminals, $t1$ and $t2$. The algorithm verifies the two concentrators, $c1$ and $c2$, assigned to them. If the concentrators have enough capacities and at least one of the concentrators is closest to the terminal that will be assigned, then the algorithm exchanges the terminals, $t1$ and $t2$ between the two concentrators, $c1$ and $c2$. The algorithm repeats this process until at least one exchange is made.

Select best bee

Only the bee with the smallest fitness will be selected to form the next population.

Assign remaining bees

In standard Bees Algorithm, the remaining bees in the population ($ns - nss$) are assigned randomly. In our implementation the scouts can be created using the Greedy Algorithm proposed by Abuali et al. [8].

Termination criteria

The algorithm stops when a maximum number of iterations (mi) is reached.

Further information on Bees Algorithm can be found in [9].

4 Results

In order to test the performance of our approach, we use a collection of TA instances of different sizes. We take 9 problems from literature [10].

The better results obtained with the Bees Algorithm use ns between 10 and 50, nss between $ns/2$ and ns , nbs between $nss/2$ and nss , $nbss$ between 30 and 100 and $nbbs$ between 30 and 100. These parameters were experimentally considered good and robust for the problems tested.

Small populations are very desirable for reducing the required computational resources. The Bees Algorithm has a good performance using initially a small population (Fig. 3).

For parameter nm , the number of modifications nm between $[N/20 \dots N/5]$ has been shown experimentally to be more efficient (Fig. 4). In our experiments nm was set to $\{0, 1, 2, \dots, N\}$. A high nm has a significant impact on the execution time (Fig. 4). A small nm did not allow the system to escape from local minima, because the resulting solution was in most cases the same as the initial permutation.

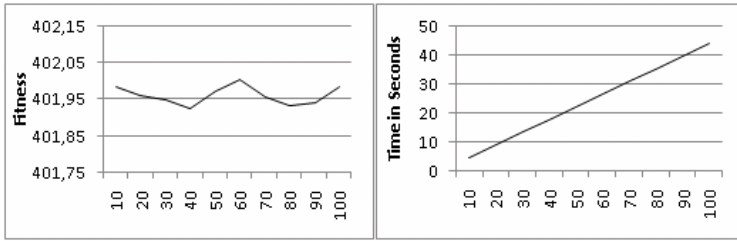


Fig. 3. Number of Bees (ns) – Average Fitness/Execution Time – Problem 7

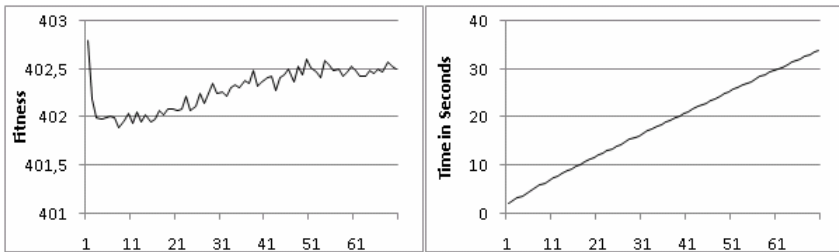


Fig. 4. Number of modifications – Average Fitness/Execution Time – Problem 7

In general, the experiments have shown that the proposed parameter setting is very robust to small modifications.

To compare our results we consider the results produced with the Local Search Genetic Algorithm, the Tabu Search Algorithm, the Hybrid Differential Evolution Algorithm and the Hybrid Ant Colony Optimization Algorithm. The GA was first applied to TA by Abuali et al. [8]. The GA is widely used in literature to make comparisons with other algorithms. TS was applied to this problem by Xu et al. [11] and Bernardino et al. [10]. We compare our algorithm with the TS, LSGA, HDE and HACO algorithms proposed by Bernardino et al. [10][12][6][13], because they use the same test instances.

Table 1 presents the best-obtained results with Bees Algorithm, TS, LSGA, HDE and HACO. The first column represents the number of the problem (Prob) and the remaining columns show the results obtained (BestF – Best Fitness, Time – Run Times) by the five algorithms. The algorithms have been executed using a processor Intel Core Duo T2300. The run time corresponds to the average time that the algorithms need to obtain the best feasible solution.

Table 2 presents the average fitnesses and standard deviations. The first column represents the number of the problem (Prob) and the remaining columns show the results obtained (AvgF – Average Fitness, Std – Standard Deviation) by the five algorithms. To compute the results in table 2 we use 300 iterations/generations for instances 1-4, 500 for instance 5, 1000 for instance 6, 1500 for instance 7 and 2000 for instances 8-9. The parameters of the Bees Algorithm are set to $n_s=10$, $n_{ss}=10$, $n_{bs}=5$, $n_{bss}=10$ and $n_{bbs}=30$ and $n_m=6$. The HDE and LSGA were applied to populations of 200 individuals. The HACO was applied to populations of 30 individuals. The initial solutions were created using the Greedy Algorithm.

Table 1. Results

| Prob | LSGA | | Tabu Search | | HDE | | HACO | | Bees | |
|------|--------|------|-------------|------|--------|------|--------|------|--------|------|
| | BestF | Time | BestF | Time | BestF | Time | BestF | Time | BestF | Time |
| 1 | 65,63 | <1s | 65,63 | <1s | 65,63 | <1s | 65,63 | <1s | 65,63 | <1s |
| 2 | 134,65 | <1s | 134,65 | <1s | 134,65 | <1s | 134,65 | <1s | 134,65 | <1s |
| 3 | 270,26 | <1s | 270,26 | <1s | 270,26 | <5s | 270,26 | <1s | 270,26 | <1s |
| 4 | 286,89 | <1s | 286,89 | <1s | 286,89 | <5s | 286,89 | <1s | 286,89 | <1s |
| 5 | 335,09 | <1s | 335,09 | <1s | 335,09 | <5s | 335,09 | 2s | 335,09 | <1s |
| 6 | 371,12 | 1s | 371,12 | <1s | 371,12 | 58s | 371,12 | 3s | 371,12 | <1s |
| 7 | 401,21 | 1s | 401,49 | 1s | 401,21 | 118s | 401,21 | 4s | 401,21 | <1s |
| 8 | 563,19 | 7s | 563,34 | 1s | 563,19 | 274s | 563,19 | 14s | 563,19 | 3s |
| 9 | 642,83 | 7s | 642,86 | 2s | 642,83 | 456s | 642,83 | 25s | 642,83 | 4s |

Table 2. Results – average fitnesses and standard deviations

| Prob | LSGA | | TS | | HDE | | HACO | | Bees | |
|------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|-------------|---------------|-------------|
| | AvgF | Std | AvgF | Std | AvgF | Std | AvgF | Std | AvgF | Std |
| 1 | 65,63 | 0,00 | 65,63 | 0,00 | 65,63 | 0,00 | 65,63 | 0,00 | 65,63 | 0,00 |
| 2 | 134,65 | 0,00 | 134,65 | 0,00 | 134,65 | 0,00 | 134,65 | 0,00 | 134,65 | 0,00 |
| 3 | 270,32 | 0,06 | 270,48 | 0,15 | 270,35 | 0,06 | 270,32 | 0,06 | 270,29 | 0,04 |
| 4 | 286,90 | 0,02 | 287,93 | 0,75 | 286,97 | 0,09 | 286,91 | 0,04 | 286,89 | 0,00 |
| 5 | 335,34 | 0,25 | 336,00 | 0,66 | 335,42 | 0,16 | 335,11 | 0,03 | 335,11 | 0,02 |
| 6 | 371,57 | 0,22 | 372,35 | 0,51 | 371,60 | 0,17 | 371,55 | 0,17 | 371,21 | 0,10 |
| 7 | 401,87 | 0,24 | 403,29 | 0,76 | 401,58 | 0,12 | 401,61 | 0,15 | 401,45 | 0,12 |
| 8 | 563,59 | 0,24 | 564,34 | 0,59 | 564,03 | 0,21 | 563,55 | 0,16 | 563,37 | 0,11 |
| 9 | 643,83 | 0,41 | 644,04 | 0,53 | 646,65 | 0,61 | 643,67 | 0,38 | 643,41 | 0,25 |

The values presented have been computed based on 50 different executions (50 best executions out of 100 executions) for each test instance.

The five algorithms reach feasible solutions for all test instances. In comparison, the Bees Algorithm presents a better average fitness for larger instances. The Bees Algorithm can reach the best-known solutions for all instances. HDE, HACO and LSGA can also find the best-known solutions, but in a higher execution time.

As it can be seen, for larger instances, the standard deviations and the average fitnesses for the Bees Algorithm are smaller. It means that the Bees Algorithm is more robust than TS, LSGA, HDE and HACO.

All the statistics obtained show that the performance of Bees Algorithm is superior to TS, LSGA, HDE and HACO.

5 Conclusions

In this paper we present a Bees Algorithm to solve the Terminal Assignment Problem. The performance of Bees Algorithm is compared with four algorithms from the literature, namely, LSGA, TS, HDE and HACO. The Bees Algorithm is a new evolutionary optimization technique, capable of performing simultaneous local and global search.

Relatively to the problem studied, the Bees Algorithm presents better results. The computational results show that Bees Algorithm had a stronger performance, improving

the results obtained by previous approaches. Moreover, in terms of standard deviation, the Bees Algorithm also proved to be more stable and robust than the other algorithms.

Experimental results demonstrate that the proposed Bees Algorithm is an effective and competitive approach in composing satisfactory results with respect to solution quality and execution time for the Terminal Assignment Problem.

In literature the application of Bees Algorithm for this problem is nonexistent, for that reason this article shows its enforceability in the resolution of this problem.

The implementation of parallel algorithms will speed up the optimization process.

References

1. Khuri, S., Chiu, T.: Heuristic Algorithms for the Terminal Assignment Problem. In: Proc. of the ACM Symposium on Applied Computing, pp. 247–251. ACM Press, New York (1997)
2. Salcedo-Sanz, S., Yao, X.: A hybrid Hopfield network-genetic algorithm approach for the terminal assignment problem. *IEEE Transaction On Systems, Man and Cybernetics*, 2343–2353 (2004)
3. Yao, X., Wang, F., Padmanabhan, K., Salcedo-Sanz, S.: Hybrid evolutionary approaches to terminal assignment in communications networks. In: *Recent Advances in Memetic Algorithms and related search technologies*, vol. 166, pp. 129–159. Springer, Berlin (2005)
4. Pham, D.T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., Zaidi, M.: The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK (2005)
5. Pham, D.T., Afify, A.A., Koç, E.: Manufacturing cell formation using the Bees Algorithm. In: *Innovative Production Machines and Systems Virtual Conference*, Cardiff, UK (2007)
6. Bernardino, E., Bernardino, A., Sánchez-Pérez, J., Vega-Rodríguez, M., Gómez-Pulido, J.: A Hybrid Differential Evolution Algorithm for solving the Terminal assignment problem. In: *International Symposium on Distributed Computing and Artificial Intelligence 2009*, pp. 178–185. Springer, Heidelberg (2009)
7. Kennedy, J., Eberhart, R.C., Shi, Y.: *Swarm intelligence*. Morgan Kaufmann, San Francisco (2001)
8. Abuali, F., Schoenefeld, D., Wainwright, R.: Terminal assignment in a Communications Network Using Genetic Algorithms. In: *Proc. of the 22nd Annual ACM Computer Science Conference*, pp. 74–81. ACM Press, New York (1994)
9. Bees Algorithm Website, <http://www.bees-algorithm.com/>
10. Bernardino, E., Bernardino, A., Sánchez-Pérez, J., Vega-Rodríguez, M., Gómez-Pulido, J.: Tabu Search vs Hybrid Genetic Algorithm to solve the terminal assignment problem. In: *IADIS International Conference Applied Computing*, pp. 404–409. IADIS Press (2008)
11. Xu, Y., Salcedo-Sanz, S., Yao, X.: Non-standard cost terminal assignment problems using tabu search approach. In: *IEEE Conference in Evolutionary Computation*, vol. 2, pp. 2302–2306 (2004)
12. Bernardino, E., Bernardino, A., Sánchez-Pérez, J., Vega-Rodríguez, M., Gómez-Pulido, J.: Solving the Terminal Assignment Problem Using a Local Search Genetic Algorithm. In: *International Symposium on Distributed Computing and Artificial Intelligence*, pp. 225–234. Springer, Heidelberg (2008)
13. Bernardino, E., Bernardino, A., Sánchez-Pérez, J., Vega-Rodríguez, M., Gómez-Pulido, J.: A Hybrid Ant Colony Optimization Algorithm for Solving the Terminal Assignment Problem. In: *International Conference on Evolutionary Computation* (2009)

Multicriteria Assignment Problem (Selection of Access Points)

Mark Sh. Levin¹ and Maxim V. Petukhov²

¹ Inst. for Information Transmission Problems,
Russian Academy of Sciences, Moscow 127994, Russia
mslevin@acm.org

² Moscow Inst. of Physics and Technology, State University,
Dolgoprudny 141700, Russia
maxim@frtk.ru

Abstract. This paper addresses assignment of users to access points of wireless network. The considered problem is based on multicriteria assignment model. A set of examined criteria involves the following: (i) maximum of bandwidth, (ii) number of users which are under service at the same time, (iii) network reliability requirements, etc. Two kinds of resource constraints are examined: (a) the number of users under service for each access point, (b) frequency bandwidth that is provided by each access point. The considered optimization problem is NP-hard and heuristic is proposed. Numerical examples illustrate the approach.

Keywords: Assignment problem, heuristics, combinatorial optimization, multicriteria decision making, wireless telecommunication network.

1 Introduction

In recent years the significance of connection between users (clients) and access points of wireless communication networks has been increased (e.g., [8], [9], [12]). This paper addresses assignment of users to wireless telecommunication network access points of a wireless network. Fig. 1 illustrates users and access points of a wireless telecommunication network.

Here the considered problem is firstly based on multicriteria generalized assignment/allocation model. Generalized assignment problems have been intensively studied (e.g., [1], [7], [16], [20]). In this article multicriteria generalized assignment problem is firstly examined with application to wireless telecommunication networks. A set of examined criteria involves the following: (i) maximum of bandwidth, (ii) number of users which are under service at the same time, (iii) network reliability requirements, etc. Multicriteria assignment problem is formulated (NP-hard [5]) and heuristic is proposed. Two problem versions are examined: (i) each user is connected to the only one access points, (ii) a user can be connected to several access points. A numerical example illustrates the approach. Authors MatLab programs (<http://www.mathworks.com/>) were used for computing.

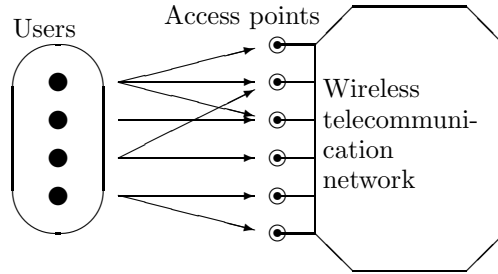


Fig. 1. Users and access points

2 Problem Statement

The following is assumed: (1) there is a hilly terrain; (2) access points of the wireless telecommunication network are distributed over the terrain; (3) users can be distributed arbitrarily over there; (4) each user requires an access to an access point of the wireless telecommunication network (i.e., assignment) and each assignment "user-access point" is described by a set of parameters; and (5) access points and users have coordinates (x, y, z) at the terrain map.

Our engineering problem consists in the following:

Assign maximal number of users to access points of wireless telecommunication network to maximize a generalized reliability of connection, quality of signal propagation, quality of usage of frequency spectrum, QoS, quality of information transmission protection while taking into account requirements of users: (a) frequency spectrum, (b) level of information transmission protection, etc.

Now let us consider a basic problem statement. Let $\Psi = \{1, \dots, i, \dots, n\}$ be a set of users and $\Theta = \{1, \dots, j, \dots, m\}$ be a set of access points. Each user i is described by parameter vector $(x_i, y_i, z_i, f_i, k_i, p_i, r_i, d_i)$, where components are as follows: coordinates of user (x_i, y_i, z_i) , parameter corresponding to required frequency bandwidth (1 Mbit/s ... 10 Mbit/s) f_i , maximal possible number of access points for connection k_i (this parameter is used for an extended 2nd problem version), required level of CoS (class, priority) p_i , required reliability of information transmission r_i , required level of information protection d_i . Each access point is described as follows: $(x_j, y_j, z_j, f_j, n_j, r_j, d_j)$, where coordinates of access point (x_j, y_j, z_j) , parameter corresponding to maximal possible traffic (i.e., maximum of possible bandwidth) f_j , maximal possible number of users under service n_j , reliability of channel for data transmission r_j , parameter of information protection d_j . Table 1 contains description of scales for parameters above.

As a result, each pair "user-access point" can be described as well: $(\forall(i, j), i \in \Psi, j \in \Theta)$ can be described as well by the following parameter: (1) level of reliability r_{ij} , (2) parameter of distance and existence of a barrier (i.e, quality of signal propagation) β_{ij} , (3) parameter of using a bandwidth f_{ij} , (4) level of QoS (class or priority) p_{ij} , (5) parameter of data protection d_{ij} . Thus, the following vector parameter is obtained $\forall(i, j), i \in \Psi, j \in \Theta: \widehat{c}_{ij} = (r_{ij}, \beta_{ij}, f_{ij}, p_{ij}, d_{ij})$.

Table 1. Parameters and scales description

| Parameter | Scale | Description |
|----------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| p_i (user) | [1, 3] | $p_i = 1$: all user requirements have to be satisfied, $p_i = 2$: half of the required frequency bandwidth may be used, corresponding users have the second level of priority, reliability can be decreased; $p_i = 3$: connection of user can be realized in the case of any bandwidth. |
| r_i (user) | [1, 10] | $r_i = 1$: information can be lost (up to 20 %, e.g., movings), $r_i = 10$: information cannot be lost (maximal reliability required by user). |
| r_j (access point) | [1, 10] | $r_j = 1$: information can be lost (up to 20 %), $r_j = 10$: information cannot be lost. |
| d_i (user) | [1, 10] | $d_i = 1$: information is not confidential, $d_i = 5$: medium level of information protection $d_i = 10$: information is confidential. |
| d_j (access point) | [1, 10] | $d_j = 1$: trivial tools for data protection, $d_j = 10$: the highest level of data protection. |

The assignment of user i to access point j is defined by Boolean variable x_{ij} ($x_{ij} = 1$ in the case of assignment i to j and $x_{ij} = 0$ otherwise). Thus, the assignment solution ($\Psi \Rightarrow \Theta$) is defined by Boolean matrix $X = ||x_{ij}||$, $i = \overline{1, n}$, $j = \overline{1, m}$.

Now let us consider computing rules for assignment user i ($\forall i \in \Psi$) to access point j ($\forall i \in \Theta$):

(1) Reliability: $r_{ij} = \min\{r_i, r_j\}$.

(2) Distance: l_{ij} .

(3) Parameter of barrier: $e_{ij} = \begin{cases} 1, & \text{barrer exists,} \\ 0, & \text{barrier is absent.} \end{cases}$

(4) Integrated parameter (barrier & distance) ($L_{max} = \max_{\{(i,j)\}} l_{ij}$):

$$\beta_{ij} = \begin{cases} 0, & (l_{ij} > L_{max}/2) \& (e_{ij} = 1), \\ 5, & (l_{ij} < L_{max}/2) \& (e_{ij} = 1) \text{ or } (l_{ij} \in (L_{max}/2, L_{max}) \& (e_{ij} = 0)), \\ 10, & (l_{ij} < L_{max}/2) \& (e_{ij} = 0). \end{cases}$$

Parameter of "connectivity" by β_{ij} is: $\xi_{ij}^\beta = \begin{cases} 0, & \text{if } \beta_{ij} = 0, \\ 1, & \text{otherwise.} \end{cases}$

(5) QoS (priority): $p_{ij} = p_i$.

(6) Required/possible bandwidth: f_{ij} (at initial stage $f_{ij} = f_i$). Three cases are examined:

(a) $p_{ij} = 1$: $f_{ij} = f_i$,

(b) $p_{ij} = 2$: $f_{ij} = \begin{cases} f_i, & \text{if } (\max_j f_j - \frac{1}{m} \sum_{j=1}^m f_j) \geq f_i, \\ \max_j f_j - \frac{1}{m} \sum_{j=1}^m f_j, & \text{if } (\max_j f_j - \frac{1}{m} \sum_{j=1}^m f_j) < f_i, \end{cases}$

$$(c) p_{ij} = 3: f_{ij} = \begin{cases} f_i, & \text{if } (\max_j f_j - \frac{1}{m} \sum_{j=1}^m f_j) \geq f_i, \\ \max_j f_j - \frac{1}{m} \sum_{j=1}^m f_j, & \text{if } (\max_j f_j - \frac{1}{m} \sum_{j=1}^m f_j) < f_i. \end{cases}$$

Here it is assumed, that two kinds of traffic exist: (i) *elastic* (users service uses only an accessible bandwidth), (ii) *non-elastic* (user has his requirements to a certain bandwidth for users services).

(7) Parameter of protection for data transmission: d_{ij} . Three cases are under examination:

$$(a) p_{ij} = 1: d_{ij} = \begin{cases} d_j, & \text{if } d_j \geq d_i, \\ 0, & \text{if } d_j < d_i; \end{cases}$$

$$(b) p_{ij} = 2: d_{ij} = \begin{cases} d_j, & \text{if } d_j \geq d_i/2, \\ 0, & \text{if } d_j < d_i/2; \end{cases}$$

$$(c) p_{ij} = 3: d_{ij} = d_j.$$

Parameter of "connectivity" by d_{ij} is: $\xi_{ij}^d = \begin{cases} 0, & \text{if } d_{ij} = 0, \\ 1, & \text{otherwise.} \end{cases}$

In addition, $\forall i \in \Psi$ is defined $\Theta_i \subseteq \Theta$. Clearly, if $|\Theta_i| = 0$, user i can be deleted from the future examination. This situation corresponds to parameters $\xi_{ij}^\beta, \xi_{ij}^d$. Fig. 2 depicts data processing.

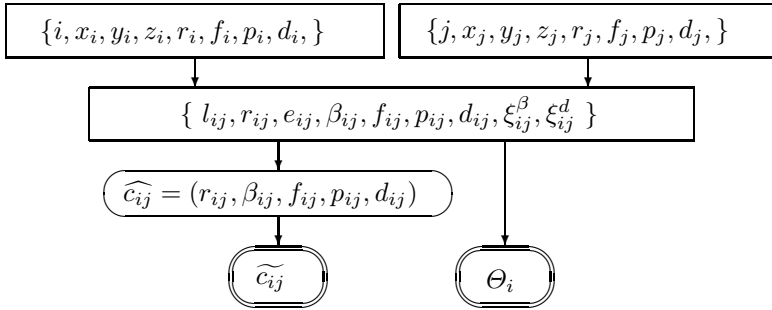


Fig. 2. Data processing

The following set of generalized objective functions (criteria) is used (a simplified additive versions): (i) total reliability $R(X) = \sum_{i=1}^n \sum_{j=1}^m r_{ij} x_{ij}$, (ii) total parameter of quality for signal propagation $B(X) = \sum_{i=1}^n \sum_{j=1}^m \beta_{ij} x_{ij}$, (iii) generalized quality of usage of frequency spectrum $F(X) = \sum_{i=1}^n \sum_{j=1}^m f_{ij} x_{ij}$, (iv) generalized parameter of QoS $P(X) = \sum_{i=1}^n \sum_{j=1}^m p_{ij} x_{ij}$, (v) generalized parameter of protection for information transmission $D(X) = \sum_{i=1}^n \sum_{j=1}^m d_{ij} x_{ij}$.

Thus, vector-like quality of solution X is:

$$\overline{C(X)} = (R(X), B(X), F(X), P(X), D(X)).$$

Further, let us consider constraints:

1. For bandwidth of access point j : $\sum_{i=1}^n f_{ij} x_{ij} \leq f_j \quad \forall j \in \Theta$, where f_j is maximum of bandwidth for access point j .

2. For number of users in each access point j : $\sum_{i=1}^n x_{ij} \leq n_j \quad \forall j \in \Theta$, where n_j is the maximum of users which are assigned to access point j .
3. For assignment of users to access point: (i) version 1 (each user is assigned to the only one access point): $\sum_{j \in \Theta_i} x_{ij} \leq 1 \quad \forall i \in \Psi$ or (ii) version 2 (a user can be assigned to several access points): $\sum_{j \in \Theta_i} x_{ij} \leq k_i \quad \forall i \in \Psi$.

It is reasonable to point out connection of a user to several access points can lead to the following: (a) transmission of data through several access points (i.e., different routes) can provide an increased reliability, (b) initial information can be divided into part which are transmitted through different access points (i.e., routes) with synthesis at a destination point and this approach can provide an increased transmission protection. Finally the problem (version 1) is:

$$\begin{aligned} \max R(X) &= \sum_{i=1}^n \sum_{j \in \Theta_i} r_{ij} x_{ij}, & \max B(X) &= \sum_{i=1}^n \sum_{j \in \Theta_i} \beta_{ij} x_{ij}, \\ \max F(X) &= \sum_{i=1}^n \sum_{j \in \Theta_i} f_{ij} x_{ij}, & \max P(X) &= \sum_{i=1}^n \sum_{j \in \Theta_i} p_{ij} x_{ij}, \\ \max D(X) &= \sum_{i=1}^n \sum_{j \in \Theta_i} d_{ij} x_{ij} \\ \text{s.t. } \sum_{i=1}^n f_{ij} x_{ij} &\leq f_j \quad \forall j \in \Theta, \quad \sum_{i=1}^n x_{ij} \leq n_j \quad \forall j \in \Theta, \quad \sum_{j \in \Theta_i} x_{ij} \leq 1 \quad \forall i \in \Psi, \\ x_{ij} &= 0 \cup 1, \quad \forall i = \overline{1, n}, \quad \forall j = \overline{1, m}, \quad x_{ij} = 0, \quad \forall i = \overline{1, n}, \quad j \in \{\Theta \setminus \Theta_i\}. \end{aligned}$$

Evidently, in version 2 another constraint 3 is used: $\sum_{j \in \Theta_i} x_{ij} \leq k_i \quad \forall i \in \Psi$.

3 Solving Scheme

The obtained combinatorial problem is NP-hard ([5], [6]). In recent decades active research projects have been conducted in the field of multicriteria assignment/ allocation (e.g., [4], [11], [14], [15], [17], [18]). Usually the following approaches are used:

- (1) enumerative methods (e.g., branch-and-bound methods) (e.g., [1], [16], [17]);
- (2) interactive (man-machine) procedures [11];
- (3) reducing an initial optimization model to a simplified problem, for example, reducing a multicriteria problem to an one-criterion problem and usage of efficient (i.e., polynomial) algorithms, e.g., Hungarian method, etc. (e.g., [10]);
- (4) heuristics including the following: (a) simple greedy algorithms, (b) approximation algorithms, (c) random algorithms, (d) meta-heuristics (e.g., hybrid algorithms), (e) variable neighborhood search VNS, (f) genetic algorithms, evolutionary multiobjective optimization (e.g., [3], [15], [19]); etc.

In this work three solving schemes were under examination:

Scheme 1. An enumerative algorithm.

Scheme 2. Two-stage heuristic: (i) simplification of the problem that was based on mapping of parameter vector for connection "user-access point" to an ordinal scale, here multicriteria ranking is used as a modification of ELECTRE technique, (ii) solving the obtained one-criterion assignment problem (e.g., greedy algorithm).

Scheme 3. Three-stage heuristic: (i) solving of an initial multicriteria problem for each criterion to generate a corresponding set of solutions, (ii) unification of the obtained solution sets and revelation of Pareto-efficient solutions, (iii) analysis of the obtained Pareto-efficient solutions and selection of the best one (or ones) (here additional rules and/or expert judgment can be used).

Further results of using Scheme 2 above are described. It is reasonable to note, the considered assignment problem may have many applications for dynamical modes of telecommunication networks (i.e., Ad Hoc networks, mobile networks, mesh networks) and usually there is a very limited time interval for the solving process. Thus, it is necessary to use simple and very efficient heuristics (e.g., greedy algorithms).

4 Numerical Example

The considered example consists of 22 users and 6 access points (Fig. 3, Fig. 4). Tables 2 and 3 contain initial information for users and access points. Vectors $(\xi_{ij}^\beta, \xi_{ij}^d)$ (Table 4) define sets $\{\Theta_i\}$ ($i = \overline{1, n}$): $\Theta_1 = \{1, 3, 5\}$, $\Theta_2 = \{1, 2, 3, 5\}$, $\Theta_3 = \{1, 2, 3, 4, 5\}$, $\Theta_4 = \{1, 2, 3, 4\}$, $\Theta_5 = \{2, 4\}$, $\Theta_6 = \{2, 4, 6\}$, $\Theta_7 = \{1, 3, 4, 5\}$, $\Theta_8 = \{1, 2, 3, 4, 5, 6\}$, $\Theta_9 = \{1, 2, 3, 4, 5, 6\}$, $\Theta_{10} = \{2, 3, 4, 5, 6\}$, $\Theta_{11} = \{2, 3, 4, 5, 6\}$, $\Theta_{12} = \{1, 3, 4, 5\}$, $\Theta_{13} = \{1, 3, 4, 5\}$, $\Theta_{14} = \{1, 3, 4, 5, 6\}$, $\Theta_{15} = \{2, 3, 4, 6\}$, $\Theta_{16} = \{1, 3, 4, 5\}$, $\Theta_{17} = \{1, 5\}$, $\Theta_{18} = \{3, 4, 5, 6\}$, $\Theta_{19} = \{2, 3, 4, 6\}$, $\Theta_{20} = \{1, 3, 4, 5, 6\}$, $\Theta_{21} = \{1, 3, 5\}$, and $\Theta_{22} = \{4, 6\}$.

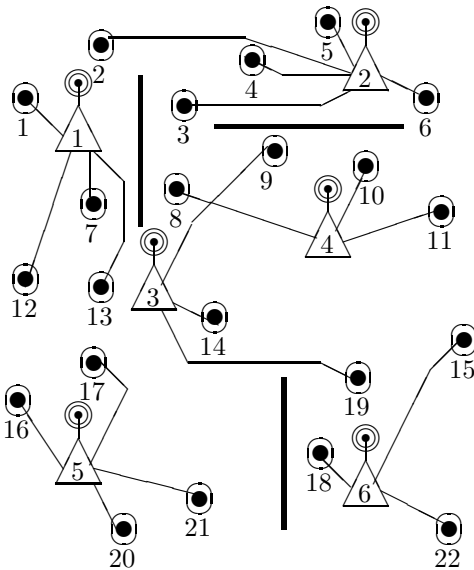


Fig. 3. Assignment of users (version 1)

Table 2. Access points

| j | x_j | y_j | z_j | f_j | n_j | r_j | d_j |
|-----|-------|-------|-------|-------|-------|-------|-------|
| 1 | 50 | 157 | 10 | 30 | 4 | 10 | 10 |
| 2 | 150 | 165 | 10 | 30 | 5 | 15 | 8 |
| 3 | 72 | 102 | 10 | 42 | 6 | 10 | 9 |
| 4 | 140 | 112 | 10 | 32 | 5 | 8 | 8 |
| 5 | 45 | 52 | 10 | 45 | 10 | 10 | 10 |
| 6 | 147 | 47 | 10 | 30 | 5 | 15 | 7 |

Computing of integrated parameters of correspondence \widetilde{c}_{ij} (Table 5) is based on mapping of vector estimate \widehat{c}_{ij} (Tables 5 and 6) into ordinal scale [1,3] (3 corresponds to the best level of correspondence, multicriteria ranking based on ELECTRE technique is used). In addition to Tables 5 and 6, sets $\{\Theta_i\}$ ($\widetilde{c}_{ij} = 0$ if $j \in \{\Theta \setminus \Theta_i\}$) are taken into account. Thus, a simplified one-criterion assignment problem is solved (version 1, i.e., $k_i = 1 \forall i = \overline{1, n}$):

$$\begin{aligned} & \max \sum_{i=1}^n \sum_{j \in \Theta_i} \widetilde{c}_{ij} x_{ij} \\ & \text{s.t. } \sum_{i=1}^n f_{ij} x_{ij} \leq \overline{f_j} \forall j \in \Theta, \sum_{i=1}^n x_{ij} \leq n_j \forall j \in \Theta, \sum_{j \in \Theta_i} x_{ij} \leq 1 \forall i \in \Psi, \\ & x_{ij} = 0 \cup 1, \forall i = \overline{1, n}, \forall j = \overline{1, m}, x_{ij} = 0, \forall i = \overline{1, n}, j \in \{\Theta \setminus \Theta_i\}. \end{aligned}$$

In version 2 another constraint 3 is used: $\sum_{j \in \Theta_i} x_{ij} \leq k_i \forall i \in \Psi$. Results of the solving process are: (i) Fig. 3 (version 1) and (ii) Fig. 4 (version 2).

Table 3. Users

| i | x_i | y_i | z_i | f_i | k_i | p_i | r_i | d_i |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | 30 | 165 | 5 | 10 | 2 | 2 | 5 | 8 |
| 2 | 58 | 174 | 5 | 5 | 1 | 1 | 9 | 6 |
| 3 | 88 | 156 | 0 | 6 | 1 | 1 | 6 | 8 |
| 4 | 110 | 169 | 5 | 7 | 1 | 2 | 5 | 6 |
| 5 | 145 | 181 | 3 | 5 | 1 | 2 | 4 | 6 |
| 6 | 170 | 161 | 5 | 7 | 1 | 2 | 4 | 7 |
| 7 | 52 | 134 | 5 | 6 | 1 | 1 | 8 | 7 |
| 8 | 85 | 134 | 3 | 6 | 1 | 1 | 7 | 7 |
| 9 | 120 | 140 | 6 | 4 | 1 | 2 | 6 | 8 |
| 10 | 150 | 136 | 3 | 6 | 1 | 2 | 7 | 8 |
| 11 | 175 | 125 | 1 | 8 | 1 | 3 | 5 | 6 |
| 12 | 27 | 109 | 7 | 8 | 1 | 3 | 5 | 8 |
| 13 | 55 | 105 | 2 | 7 | 1 | 2 | 10 | 6 |
| 14 | 98 | 89 | 3 | 10 | 3 | 1 | 10 | 7 |
| 15 | 183 | 91 | 4 | 4 | 1 | 3 | 5 | 7 |
| 16 | 25 | 65 | 2 | 7 | 1 | 3 | 5 | 6 |
| 17 | 52 | 81 | 1 | 10 | 3 | 1 | 8 | 10 |
| 18 | 135 | 59 | 4 | 13 | 1 | 3 | 4 | 6 |
| 19 | 147 | 79 | 5 | 7 | 1 | 3 | 16 | 8 |
| 20 | 65 | 25 | 7 | 6 | 1 | 2 | 9 | 8 |
| 21 | 93 | 39 | 1 | 10 | 2 | 1 | 10 | 9 |
| 22 | 172 | 26 | 2 | 10 | 1 | 2 | 7 | 6 |

Table 4. Matrix $\|(\xi_{ij}^\beta, \xi_{ij}^d)\|$

| i | Access points j | | | | | |
|-----|-------------------|-----|-----|-----|-----|-----|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 1,1 | 0,1 | 1,1 | 0,1 | 1,1 | 0,1 |
| 2 | 1,1 | 1,1 | 1,1 | 0,1 | 1,1 | 0,1 |
| 3 | 1,1 | 1,1 | 1,1 | 1,1 | 1,1 | 0,0 |
| 4 | 1,1 | 1,1 | 1,1 | 1,1 | 0,1 | 0,1 |
| 5 | 0,1 | 1,1 | 0,1 | 1,1 | 0,1 | 0,1 |
| 6 | 0,1 | 1,1 | 0,1 | 1,1 | 0,1 | 1,1 |
| 7 | 1,1 | 0,1 | 1,1 | 1,1 | 1,1 | 0,1 |
| 8 | 1,1 | 1,1 | 1,1 | 1,1 | 1,1 | 1,1 |
| 9 | 1,1 | 1,1 | 1,1 | 1,1 | 1,1 | 1,1 |
| 10 | 0,1 | 1,1 | 1,1 | 1,1 | 1,1 | 1,1 |
| 11 | 0,1 | 1,1 | 1,1 | 1,1 | 1,1 | 1,1 |
| 12 | 1,1 | 0,1 | 1,1 | 1,1 | 1,1 | 0,1 |
| 13 | 1,1 | 0,1 | 1,1 | 1,1 | 1,1 | 0,1 |
| 14 | 1,1 | 0,1 | 1,1 | 1,1 | 1,1 | 1,1 |
| 15 | 0,1 | 1,1 | 1,1 | 1,1 | 0,1 | 1,1 |
| 16 | 1,1 | 0,1 | 1,1 | 1,1 | 1,1 | 0,1 |
| 17 | 1,1 | 0,0 | 1,0 | 1,0 | 1,1 | 0,0 |
| 18 | 0,1 | 0,1 | 1,1 | 1,1 | 1,1 | 1,1 |
| 19 | 0,1 | 1,1 | 1,1 | 1,1 | 0,1 | 1,1 |
| 20 | 1,1 | 0,1 | 1,1 | 1,1 | 1,1 | 1,1 |
| 21 | 1,1 | 0,0 | 1,1 | 1,0 | 1,1 | 1,0 |
| 22 | 0,1 | 0,1 | 0,1 | 1,1 | 0,1 | 1,1 |

Table 5a. Matrix $\|\widehat{c}_{ij}\| = \|(r_{ij}, \beta_{ij}, f_{ij}, p_{ij}, d_{ij})\|$ (part 1)

| <i>i</i> | Access points <i>j</i> | | |
|----------|------------------------|-----------------|------------------|
| | 1 | 2 | 3 |
| 1 | 5, 10, 10, 2, 10 | 5, 0, 10, 2, 8 | 5, 10, 10, 2, 9 |
| 2 | 9, 10, 5, 1, 10 | 9, 5, 5, 1, 8 | 9, 5, 5, 1, 9 |
| 3 | 6, 5, 6, 1, 10 | 6, 10, 6, 1, 8 | 6, 10, 6, 1, 9 |
| 4 | 5, 5, 7, 2, 10 | 5, 10, 7, 2, 8 | 5, 5, 7, 2, 9 |
| 5 | 4, 0, 5, 2, 10 | 4, 10, 5, 2, 8 | 4, 0, 5, 2, 9 |
| 6 | 4, 0, 7, 2, 10 | 4, 10, 7, 2, 8 | 4, 0, 7, 2, 9 |
| 7 | 8, 10, 6, 1, 10 | 8, 0, 6, 1, 8 | 8, 10, 6, 1, 9 |
| 8 | 7, 5, 6, 1, 10 | 7, 5, 6, 1, 8 | 7, 10, 6, 1, 9 |
| 9 | 6, 5, 4, 2, 10 | 6, 5, 4, 2, 8 | 6, 10, 4, 2, 9 |
| 10 | 7, 0, 6, 2, 10 | 7, 5, 6, 2, 8 | 7, 5, 6, 2, 9 |
| 11 | 5, 0, 8, 3, 10 | 5, 5, 8, 3, 8 | 5, 5, 8, 3, 9 |
| 12 | 5, 10, 8, 3, 10 | 5, 0, 8, 3, 8 | 5, 10, 8, 3, 9 |
| 13 | 10, 10, 7, 2, 10 | 10, 0, 7, 2, 8 | 10, 10, 7, 2, 9 |
| 14 | 10, 5, 10, 1, 10 | 10, 0, 10, 1, 8 | 10, 10, 10, 1, 9 |
| 15 | 5, 0, 4, 3, 10 | 5, 5, 4, 3, 8 | 5, 5, 4, 3, 9 |
| 16 | 5, 5, 7, 3, 10 | 5, 0, 7, 3, 8 | 5, 10, 7, 3, 9 |
| 17 | 8, 10, 10, 1, 10 | 8, 0, 10, 1, 0 | 8, 10, 10, 1, 0 |
| 18 | 4, 0, 10, 3, 10 | 4, 0, 10, 3, 8 | 4, 5, 10, 3, 9 |
| 19 | 10, 0, 7, 3, 10 | 15, 5, 7, 3, 8 | 10, 10, 7, 3, 9 |
| 20 | 9, 5, 6, 2, 10 | 9, 0, 6, 2, 8 | 9, 10, 6, 2, 9 |
| 21 | 10, 5, 10, 1, 10 | 10, 0, 10, 1, 0 | 10, 10, 10, 1, 9 |
| 22 | 7, 0, 10, 2, 10 | 7, 0, 10, 2, 8 | 7, 0, 10, 2, 9 |

Table 5b. Matrix $\|\widehat{c}_{ij}\| = \|(r_{ij}, \beta_{ij}, f_{ij}, p_{ij}, d_{ij})\|$ (part 2)

| i | Access points j | | |
|-----|-------------------|-------------------|-----------------|
| | 4 | 5 | 6 |
| 1 | 5, 0, 10, 2, 8 | 5, 5, 10, 2, 10 | 5, 0, 10, 2, 7 |
| 2 | 8, 0, 5, 1, 8 | 9, 5, 5, 1, 10 | 9, 0, 5, 1, 7 |
| 3 | 6, 10, 6, 1, 8 | 6, 5, 6, 1, 10 | 6, 0, 6, 1, 0 |
| 4 | 5, 5, 7, 2, 8 | 5, 0, 7, 2, 10 | 5, 0, 7, 2, 7 |
| 5 | 4, 5, 5, 2, 8 | 4, 0, 5, 2, 10 | 4, 0, 5, 2, 7 |
| 6 | 4, 10, 7, 2, 8 | 4, 0, 7, 2, 10 | 4, 5, 7, 2, 7 |
| 7 | 8, 5, 6, 1, 8 | 8, 10, 6, 1, 10 | 8, 0, 6, 1, 7 |
| 8 | 7, 10, 6, 1, 8 | 7, 10, 6, 1, 10 | 7, 0, 6, 1, 7 |
| 9 | 6, 10, 4, 2, 8 | 6, 5, 4, 2, 10 | 6, 5, 4, 2, 7 |
| 10 | 7, 10, 6, 2, 8 | 7, 5, 6, 2, 10 | 7, 10, 6, 2, 7 |
| 11 | 5, 10, 8, 3, 8 | 5, 5, 8, 3, 10 | 5, 10, 8, 3, 7 |
| 12 | 5, 5, 8, 3, 8 | 5, 10, 8, 3, 10 | 5, 0, 8, 3, 7 |
| 13 | 8, 10, 7, 2, 8 | 10, 10, 7, 2, 10 | 10, 0, 7, 2, 7 |
| 14 | 8, 10, 10, 1, 8 | 10, 10, 10, 1, 10 | 10, 5, 10, 1, 7 |
| 15 | 5, 10, 4, 3, 8 | 5, 0, 4, 3, 10 | 5, 10, 4, 3, 7 |
| 16 | 5, 5, 7, 3, 8 | 5, 10, 7, 3, 10 | 5, 0, 7, 3, 7 |
| 17 | 8, 5, 10, 1, 0 | 8, 10, 10, 1, 10 | 8, 0, 10, 1, 0 |
| 18 | 4, 10, 10, 3, 8 | 4, 5, 10, 3, 10 | 4, 10, 10, 3, 7 |
| 19 | 8, 10, 7, 3, 8 | 10, 0, 7, 3, 10 | 15, 10, 7, 3, 7 |
| 20 | 8, 5, 6, 2, 8 | 9, 10, 6, 2, 10 | 9, 5, 6, 2, 7 |
| 21 | 8, 10, 10, 1, 0 | 10, 10, 10, 1, 10 | 10, 5, 10, 1, 0 |
| 22 | 7, 10, 10, 2, 8 | 7, 0, 10, 2, 10 | 7, 10, 10, 2, 7 |

Table 6. Matrix $\|\widetilde{c}_{ij}\|$

| i | Access points j | | | | | |
|-----|-------------------|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 3 | 0 | 3 | 0 | 3 | 0 |
| 2 | 2 | 1 | 1 | 0 | 2 | 0 |
| 3 | 1 | 1 | 1 | 1 | 1 | 0 |
| 4 | 1 | 1 | 2 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 0 |
| 6 | 0 | 1 | 0 | 1 | 0 | 1 |
| 7 | 2 | 0 | 2 | 1 | 2 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 2 | 1 | 1 | 1 | 1 | 1 |
| 10 | 0 | 1 | 2 | 1 | 3 | 1 |
| 11 | 0 | 3 | 3 | 3 | 3 | 2 |
| 12 | 3 | 0 | 3 | 3 | 3 | 0 |
| 13 | 3 | 0 | 3 | 3 | 3 | 0 |
| 14 | 3 | 0 | 3 | 3 | 3 | 2 |
| 15 | 0 | 1 | 2 | 1 | 0 | 1 |
| 16 | 3 | 0 | 3 | 2 | 1 | 0 |
| 17 | 3 | 0 | 0 | 0 | 3 | 0 |
| 18 | 0 | 0 | 3 | 3 | 3 | 2 |
| 19 | 0 | 3 | 3 | 3 | 0 | 3 |
| 20 | 3 | 0 | 3 | 3 | 3 | 2 |
| 21 | 3 | 0 | 3 | 0 | 3 | 0 |
| 22 | 0 | 0 | 0 | 3 | 0 | 3 |

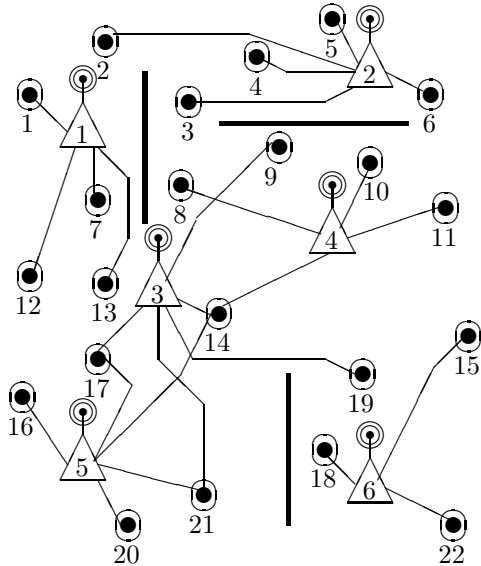


Fig. 4. Assignment of users (version 2)

5 Conclusion

The suggested approach is the first step for using multicriteria combinatorial problems in assignment of users to access points of wireless telecommunication networks. Our main attention was targeted to a new problem formulation and a simple solving scheme. Clearly the considered approach may be applied in other domains for connection of users with service centers (e.g., electricity systems, maintenance in manufacturing, environmental monitoring). It is reasonable to point out the following perspective research directions:

1. Problem statement: (1.1) problem parts: (a) procedures for computing of parameters, for example: integrated parameters (barrier-distance β_{ij}), quality of signal propagation (f_{ij}); (b) criteria, (c) constraints; (1.2) possibility to use other management modes, for example, multi-hop schemes, P2P; (1.3) mobility of access points (and users); and (1.4) on-line extension of users set.

2. Models: (2.1) more complicated optimization models (e.g., under uncertainty); (2.2) taking into account "neighbor" assignments (possible collisions, influence), here quadratic assignment problem [2] or an approach on the basis of hierarchical morphological design [14] can be used.

3. Solving methods: implementation of various solving schemes and comparison studies for different schemes.

The preliminary material for the article was prepared within framework of a faculty course "Design of Systems" in Moscow Institute of Physics and Technology (State University) (creator and lecturer: M.Sh. Levin) [13] as laboratory work 9 (student: M.V. Petukhov) and BS-thesis of M.V. Petukhov (2008, advisor: M.Sh. Levin).

References

1. Cattrysse, D.G., Van Wassenhove, L.N.: A survey of algorithms for the generalized assignment problem. *EJOR* 60(3), 260–272 (1992)
2. Cela, E.: *The Quadratic Assignment Problem*. Kluwer, Dordrecht (1998)
3. Coello, C.A.C., Van Veldhuizen, D.A., Lamont, G.B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer, Dordrecht (2002)
4. Current, J., Min, H., Schilling, D.: Multiobjective analysis of facility location decisions. *EJOR* 49(3), 295–300 (1990)
5. Fisher, M.L., Jaikumar, R., Van Wassenhove, L.: A multiplier adjustment method for the generalized assignment problem. *Manag. Sci.* 32(9), 1095–1103 (1986)
6. Garey, M.R., Johnson, D.S.: *Computers and Intractability. The Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, San Francisco (1979)
7. Gavish, B., Pirkul, H.: Algorithms for the multi-resource generalized assignment problem. *Manag. Sci.* 37(6), 695–713 (1991)
8. Jayant, N. (ed.): *Broadband Last Mile: Access Technologies for Multimedia Communications*. CRC Press/Taylor&Francis, London (2005)
9. Koutsopoulos, I., Tassiulas, L.: Joint optimal access point selection and channel assignment in wireless networks. *IEEE/ACM Trans. on Networking* 15(3), 521–532 (2007)
10. Kuhn, H.W.: The Hungarian method for the assignment problems. *Nav. Res. Log.* 52(1), 7–21 (2005)
11. Larichev, O.I., Sternin, M.: Man-computer approaches to the multicriteria assignment problem. *Autom.&Remote Control* 59(7), 135–155 (1998)
12. Levin, M.S.: *Composite Systems Decisions*. Springer, New York (2006)
13. Levin, M.S.: Student research projects in system design. In: *Proc. of 1st Int. Conf. CSEDU 2009*, Lisbon, pp. 291–295 (2009)
14. Levin, M.S.: Combinatorial optimization in system configuration design. *Automation & Remote Control* 70(3), 519–561 (2009)
15. Liang, Y.-C., Lo, M.-H.: Multi-objective redundancy allocation optimization using a variable neighborhood search algorithm. *J. of Heuristics* (2010) (in press)
16. Oncan, T.: A survey on the generalized assignment problem. *INFOR* 45(3), 123–141 (2007)
17. Orgyczak, W., Wierzbicki, A., Milewski, M.: On multi-criteria approach to fair and efficient bandwidth allocation. *Omega* 36(3), 451–463 (2008)
18. Scarelli, A., Narula, S.C.: A multicriteria assignment problem. *J. of Multi-Criteria Anal.* 11(2), 65–74 (2002)
19. Tuytens, D., Teghem, J., Fortemps, P., Van Nieuwenhuyze, K.: Performance of the MOSA method for the bicriteria assignment problem. *J. of Heuristics* 6(3), 295–310 (2000)
20. Wilson, J.M.: An algorithm for the generalized assignment problem with special ordered sets. *J. of Heuristics* 11(4), 337–350 (2005)

Composite Laminates Buckling Optimization through Lévy Based Ant Colony Optimization

Roberto Candela¹, Giulio Cottone², Giuseppe Fileccia Scimemi²,
and Eleonora Riva Sanseverino¹

¹ University of Palermo, DIEET, Viale delle Scienze Palermo, Italy

² University of Palermo, DISAG, Viale delle Scienze Palermo, Italy

Abstract. In this paper, the authors propose the use of the Lévy probability distribution as leading mechanism for solutions differentiation in an efficient and bio-inspired optimization algorithm, ant colony optimization in continuous domains, ACOR. In the classical ACOR, new solutions are constructed starting from one solution, selected from an archive, where Gaussian distribution is used for parameter diversification. In the proposed approach, the Lévy probability distributions are properly introduced in the solution construction step, in order to couple the ACOR algorithm with the exploration properties of the Lévy distribution. The proposed approach has been tested on mathematical test functions and on a real world problem of structural engineering, the composite laminates buckling load maximization. In the latter case, as in many other cases in real world problems, the function to be optimized is multi-modal, and thus the exploration ability of the Levy perturbation operator allow the attainment of better results.

1 Introduction

Recently, a lot of work has been carried out in bio-inspired computational optimization, especially in continuous domains. Among the methods set up, we can cite Evolutionary computation [1], [2] and nature inspired methods such as Ant Colony Optimization, ACO. The latter was initially developed for combinatorial optimization [3], [4] and has been recently adapted to continuous optimization [5]. Ant Colony Optimization is inspired by the ants foraging behavior and requires that the problem is partitioned into a finite set of components these being intermediate targets before reaching the ultimate goal. The solution is generally the minimum-cost strategy followed by the agent (ant) to reach the target. In Ant Colony Optimization for continuous optimization, ACOR, the partition of the problem into finite set is given by the intrinsic search space decomposition into the different dimensions.

ACOR is a population based algorithm, therefore, for each ant two main steps must be performed: the ant based solution construction and the pheromone update. The first step comprises a number of sub-operations such as: for each search space dimension: a) probabilistic choice of one base solution from an archive of

best-so-far solutions; b) perturbation of the relevant parameter following a gaussian probability distribution. The second step simply consists in the archive update. Indeed, the Gaussian distribution used for the perturbation of each parameter is built using the information derived from the entire archive. The closer the solutions are in a given dimension, the smaller the standard deviation. With respect to this issue, the authors have noticed in some cases a limited ability to perform exploration. This is especially true when multimodal functions must be minimized. Other authors [6], [7] have already observed this behavior with evolutionary programming. In particular, Lee and Yao [6] propose an adaptive LEP, Lévy based Evolutionary Programming algorithm in which the Gaussian mutation is replaced by a Lévy distributed mutation. In this paper the authors propose to modify the ACOR algorithm by replacing the Gaussian mutation with a Lévy distributed mutation, called ACOR_L. The effects are similar to those attained in [7], although different performances are observed in certain cases. The application section reports tests over a set of test functions taken from [6] as well as a real world application in the field of composite laminates buckling load maximization. Composite laminates are used in many fields of engineering due to their outstanding mechanical and structural characteristics: low weight, high stiffness and strength. Furthermore the design of a laminate could be easily accomplished changing stacking sequence, fiber orientation, ply thickness and the material used by means of standard industrial processes. Design optimization of composite structures usually leads to multimodal search spaces and different approaches were adopted to deal with this optimization problem, among others gradient based methods [8], genetic algorithms [9], simulated annealing [10], genetic algorithm and pattern search algorithm [11].

2 The Lévy Probability Distribution

In the field of global optimization, various phenomena have been already studied in the literature such as thermodynamics and evolution in the eighties. More recently, one of the laws governing enhanced diffusive processes called Lévy flights has been considered for modeling the perturbation mechanism in global optimization. In this paper, the Lévy distribution is considered for generating step size during the ACOR search. This distribution has the property of generating points that can be far from the starting ones. This can be better understood considering that Gaussian white noises random variables are symmetric about their mean and do not allow any skewness of the distribution or unilateral random input. The only distribution that allows such a great variability and obeys to a generalized central limit theorem is the so called α -stable Lévy distribution, introduced by the mathematician Paul Lévy about 1920 [12]. Stable distributions are characterized by heavy-tailed probability density function that causes infinite variance and are defined by four coefficients [13]. A random variable X is said to have a α -stable distribution if there are parameters $0 < \alpha \leq 2$, $\sigma > 0$, $1 \leq \beta \leq 1$, $\mu \in \mathbb{R}$ such that its characteristic function $\phi_X(\theta)$ has the form:

$$\phi_X(\theta) = \begin{cases} \exp\{-\sigma^\alpha|\theta|^\alpha (1 - i\beta(\text{sign}(\theta))\tan\frac{\pi\alpha}{2}) + i\mu\theta\}, & \text{if } \alpha \neq 1 \\ \exp\{-\sigma|\theta| (1 + i\beta\frac{2}{\pi}(\text{sign}(\theta))\ln|\theta|) + i\mu\theta\}, & \text{if } \alpha = 1 \end{cases} \quad (1)$$

The four parameters affect the shape of the distribution in an essential way and it is common to introduce an appropriate notation to take them into account. We denote with the symbol $X \sim S_\alpha(\sigma, \beta, \mu)$ a stable random variable with assigned parameters characterizing (1). Some properties of the stable distribution, not proved but straightforward from the definition of the characteristic function, will help to better clarify their meaning.

Addition of constant. Let $X \sim S_\alpha(\sigma, \beta, \mu)$ and let a be a real parameter. Then, adding a to X gives a random variable $X+a$ with distribution $X \sim S_\alpha(\sigma, \beta, \mu+a)$. The parameter μ is thus a shift parameter. The parameter μ cannot be identified in general as the mean of the distribution, because for $0 < \alpha < 1$, the mean of the variable $X \sim S_\alpha(\sigma, \beta, \mu)$ diverges. Only in the interval $1 < \alpha \leq 2$, the two concepts actually coincide.

Multiplication by a constant. Let be $X \sim S_\alpha(\sigma, \beta, \mu)$ and a real. Then, multiplying X by a gives a random variable aX with distribution $X \sim S_\alpha(|a|\sigma, \text{Sign}(a)\beta, a\mu)$ if $\alpha \neq 1$ and $X \sim S_\alpha(|a|\sigma, \text{Sign}(a)\beta, a\mu - (2/\pi)a(\text{Log}|a|)\sigma\beta)$ if $\alpha = 1$. σ is called scale parameter. When $\alpha = 2$, the characteristic function (1) becomes the characteristic function of a random variable normal distributed with mean μ and variance $2\sigma^2$, indicated as $X \sim N(\mu, \sqrt{2}\sigma)$. In general, the scale parameter does not coincide with the standard deviation, that, for $0 < \alpha < 2$ is infinite.

In Figure 1, two trajectories following the normal ($\alpha=2$) and the Lévy distribution ($\alpha=1.6$) are reported. The trajectories are generated by adding a Lévy distributed quantity having zero mean and $\sigma=1$ to x_1 and x_2 .

The normal path in panel (a) is sample continuous (similar, but not to be confused with the continuity of a function, and descending from the application of the Kolmogorov criterion [14] while in panel (b) the Lévy path is not sample continuous as long jumps and clustered small fluctuations are present alternate

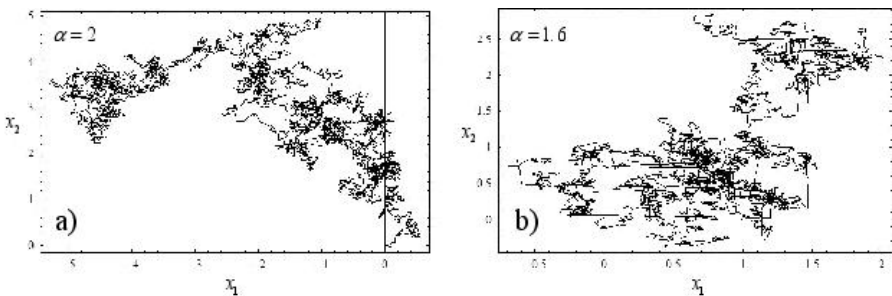


Fig. 1. Trajectories of Lévy motion: (a) typical normal path $\alpha = 2$; (b) competition between jumps and small fluctuation at $\alpha = 1.6$

with clustered small fluctuation. This is a consequence of the heavy tails of the Lévy distribution and it is influenced by the stability index: indeed, if α goes to zero, jumps become bigger and fluctuations vanish; conversely, if $\alpha=2$ the continuous (no jumps) normal behaviour is attained. Then, loosely speaking, we could say that Lévy paths tend to escape from a bounded region, while normal paths localize.

3 Function Optimization Using ACOR_L

As said in the introduction, Ant Colony Optimization was first proposed for combinatorial optimization problems. Since its emergence many attempts have been made to use it for tackling continuous problems. More recently, M.Dorigo and K. Socha [5] have proposed the natural extension of the ACO algorithm to continuous domains, ACOR_L. The idea that is central to the way ACOR works is the incremental construction of solutions based on the biased (by pheromone) probabilistic choice of solution components. At each construction step, the ant chooses a Probability Density Function. Details about the ACOR implementation are out of the scope of this paper, for further details please refer to [5]. In what follows, the main steps of the ACOR_L algorithm are briefly outlined.

Create an archive T of k solutions, $T = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^k\}$. Where $\mathbf{x}^r = [x_1^r, x_2^r, \dots, x_N^r]$. Order the solutions of the archive T according to their objective function value. Given a decision variable $x_i, i=1, \dots, N$, an ant constructs a solution by performing N construction steps. At construction step i, the ant chooses a value for the variable x_i . At this construction step, only the information related to the i-th dimension is used. Select a base solution r from the archive T to be modified according to the following probability:

$$p_r = \frac{\omega_r}{\sum_{j=1}^k \omega_j} \tag{2}$$

where

$$\omega_r = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(r-1)^2}{2q^2k^2}} \tag{3}$$

which essentially defines the weight ω_r to be a value of the Gaussian function with argument r, mean 1 and standard deviation qk, where q is a parameter of the algorithm. When q is small, the best-ranked solutions are strongly preferred, and when it is large, the probability becomes less dependent on the rank of the solution.

All the components x_i^r for $i=1$ to N of the chosen r-th solution in the following steps are perturbed following the Lévy distribution. As already pointed out, the Lévy distribution is characterised by four parameters: the scale parameter, σ , the skewness parameter, β , the shift parameter μ and the α parameter.

The first is defined as:

$$\sigma_i^r = \xi \sum_{e=1}^k \frac{|x_i^e - x_i^r|}{k-1} real \tag{4}$$

where ξ is a parameter user-defined in the algorithm ranging from 0 and 1. The higher the value of this parameter the slower the convergence speed.

The third parameter, μ , is the value of the i -th parameter of the base solution itself (x_i^r). The second parameter is set to 0, namely no dissymmetry of the probability density function about the shift value μ . The fourth parameter α is a control parameter set by the user and its value ranges between 0 and 2. So the i -th parameter is newly determined. The same procedure is repeated for all the N parameters. At the end, once the solution is entirely constructed, it is evaluated and if better than any of the solutions in T , it is included into the archive set T . From what was said above, if the used defined parameter α is set to 2, the $ACOR_L$ coincides with $ACOR$. The proposed algorithm, $ACOR_L$, for function optimization works as follow.

Algorithm 1. $ACOR_L$ Pseudocode

```

Random creation of the solutions archive of size k
Choice of  $\xi, q, \alpha, \mu$ 
while not(termination) do
  for  $z=1$  to  $m$  do
    Choice of one solution from the archive using (2)
    for all parameter (Ant construction) do
      Calculate standard deviation  $\sigma_i^z$  using (4)
      Modify the  $i$ -th parameter in the following way:
       $x^i = x^i + S_\alpha(\sigma_i^z, 0, 0)$ 
    end for
    Evaluation of the new solution
  end for
  Archive update
end while

```

4 Experimental Results and Analysis

4.1 Mathematical Test Functions

Some applications have been carried out on a test suite of 5 mathematical test functions taken from [6]. In all cases, the objective functions have to be minimized.

Except than f_1 that has no local minima, all the other functions have several local minima (f_5, f_6, f_7) or some local minima (f_{11}). In order to compare the attained results with those attained using classical $ACOR$, for both algorithms the following parameters values taken from [5] have been chosen: parameter $\xi=0.85$; parameter $q=0.0001$; archive size: 50; number of ants: 2. Parameter α has been set to different values in different runs, in order to assess its influence on the efficiency of the algorithm. Thus the following values of α have been taken 2.0, 1.8, 1.6, 1.4, 1.2, 1.0, 0.8. The termination condition is based on the maximum number of function evaluations, which has been set to 150000 for f_1, f_5, f_6, f_7 , and 3000 for f_{11} . Table 1 reports the functions f_1, f_5, f_6, f_7 and f_{11} [6]. Table 2 shows the experimental results of a comparison between the performance

Table 1. Benchmark functions used in this study. N stands for the dimension of the functions and s their ranges.

| Test function | N s |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| $f_1 = \sum_{i=1}^N x_i^2$ | 30 [-100, 100] ^N |
| $f_5 = \sum_{i=1}^N [x_i^2 - 10\cos(2\pi x_i) + 10]$ | 30 [-5.12, 5.12] ^N |
| $f_6 = -20\exp[-0.2\sqrt{\frac{1}{N}\sum_{i=1}^N x_i^2}] - \exp[\frac{1}{N}\sum_{i=1}^N \cos(2\pi x_i)]$ | 30 [-32, 32] ^N |
| $f_7 = \frac{1}{4000}\sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos(\frac{x_i}{\sqrt{i}}) + 1$ | 30 [-600, 600] ^N |
| $f_{11} = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2))x_2^2$ $(30 + (2x_1 - 3x_2)^2)(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$ | 2 [-2, 2] ^N |

Table 2. Experimental results, mean of the best solution over 100 independent runs, from standard ACOR and ACOR_L. The number in the parentheses indicate standard deviations. The last row shows the t-test. The asterisk indicates that the difference is not negligible.

| | f ₁ | f ₅ | f ₆ | f ₇ | f ₁₁ |
|-------------------|------------------|-----------------|-----------------|-----------------|-----------------|
| ACOR _L | 2.36E-144 | 4.63E+01 | 5.63E-01 | 8.93E-03 | 4.62 |
| (α=1.8) | (2.34E-143) | (1.26E+01) | (1.23) | (1.07E-02) | (1.14E+01) |
| ACOR | 2.24E-203 | 6.17E+01 | 2.28 | 2.02E-02 | 3.27 |
| (α=2.0) | (0) | (1.68E+01) | (1.38) | (2.71E-02) | (2.7) |
| t-test | 1.08 | -7.73* | -9.25* | -3.88* | 1.15 |

of ACOR and ACOR_L with the above parameters and with α=1.8. Other values of α have shown a worst behaviour.

It is clear from the table above that ACOR_L performed no worse or better in a statistically meaningful sense than ACOR on all benchmark functions having local minima. This is clearly shown by the values of the t-test.

Fig.2 shows the optimization processes for ACOR_L (ACOR is ACOR_L with α=2) with different values of α over function f₅, each point represents the mean or the standard deviation of the best so far solution at each iteration for a sample of 100 independent runs. The acronym FES stays for function evaluations.

After a fast descent of the mean, ACOR gets stuck into local minima and does not improve its performance compared to α=1.6 and α=1.8. These both improve their mean value till the very end of the run, reaching lower values of the objective function. The behaviour is confirmed looking at the standard deviation. In ACOR, the standard deviation first increases allowing a wide exploration of the search space, then decreases and after about 30000 evaluations it does not change anymore. At the same time, it can be observed that the mean value does not move anymore. For different values of α, and in particular for α=1.8, the standard deviation decreases gradually, while the search process approaches the minimum.

The behaviour of the standard deviation observed for α=2 shows a limited tendency to diversification of the attainable solutions. Higher diversification can be observed for α=1.8 and α=1.6 for a longer part of the process, leading to

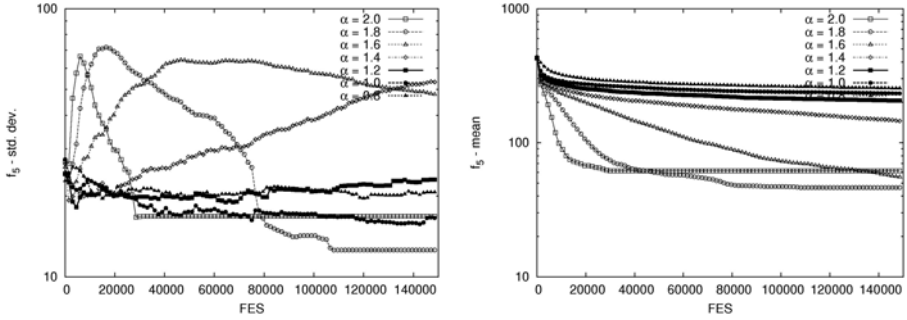


Fig. 2. Optimization process of ACOR_L over test function f_5 . Standard deviation and mean.

lower values of the mean and of the median at the end of the process. Lower median means that a high diversification has brought a large number of very good results and a limited number of bad results over the set of independent runs. A similar behaviour has been observed for the other functions showing many local minima (f_6 and f_7).

4.2 Composite Laminates Design Optimization

When a plate is subjected to in-plane compressive loads exists a value of the loads for which the originally flat equilibrium state is no longer stable. This load is called the buckling load. Before reaching the buckling load the plate has only in plane forces and deformations, membrane prebuckling state. Reaching the buckling load the plate suddenly leaves the flat state and large out of plane displacements arises usually leading to the structural collapse. The value of the buckling load depends on geometry, boundary conditions, material properties and the buckling mode shape. Considering a rectangular composite plate simply supported and subjected only to normal compressive loads the plate buckles into m and n half waves in the x and y direction, respectively, when the loads reach the values $\lambda_b N_x$ and $\lambda_b N_y$.

In the general case of laminate with multiple anisotropic layers and without any stacking sequence symmetry the problem doesn't admit a simple solution. If we assume particular constraints on the stacking sequences, i.e. plates for which the bending twisting coefficients are zero are so small in respect to the other coefficients to be assumed zero, using the classical laminate theories [15] the buckling load factor λ_b could be found as:

$$\lambda_b(m, n) = \frac{\pi^2 m^4 D_{11} + 2(D_{12} + 2D_{66})r^2 m^2 n^2 + r^4 n^4 D_{22}}{a^2 (m^2 N_x + r^2 n^2 N_y)} \quad (5)$$

where a and b are the lamina dimensions; $r = \frac{a}{b}$ the aspect ratio; N_x and N_y the applied loads; D_{ij} the bending stiffness of the composite plate depending from the assumed stacking sequence of the laminate.

The smallest value of λ_b over all possible values of m and n represents the lowest value of loads for which the buckling conditions are reached and hence the critical buckling load factor λ_{cb} . According to [9] limiting the values of m and n to 1,2 gives a good estimation of critical buckling load, so for an assigned plate geometry the optimization problem could be stated as:

$$\max_{D_{ij}} \left(\min_{m,n} \lambda_b(m, n); \quad m, n \in 1, 2 \right) \tag{6}$$

According to the classical laminate theories [15] before the buckling condition is reached the plane stress condition is assumed valid for each ply of the laminate. In the generic lamina k the constitutive equations could be expressed as:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{13} \\ \bar{Q}_{21} & \bar{Q}_{22} & \bar{Q}_{23} \\ \bar{Q}_{31} & \bar{Q}_{32} & \bar{Q}_{33} \end{bmatrix}_k \cdot \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix} \tag{7}$$

where \bar{Q}_{ij} are the lamina stiffness components expressed in the plate reference axis. The bending stiffness D_{ij} of a plate made by n lamina could be now expressed as

$$D_{ij} = \frac{1}{3} \sum_{k=1}^n \bar{Q}_{ij} (z_k^3 - z_{k-1}^3) \tag{8}$$

where z_k and z_{k-1} are the coordinate of the k lamina through the laminate thickness.

The terms \bar{Q}_{ij} could be expressed knowing the fiber orientations θ_k and the elastic properties of the material along the principal directions $E_{11}^k, E_{22}^k, G_{12}^k, \nu_{12}^k$ of each lamina, [15]. For an assumed plate geometry the design variables are hence the elastic properties and the fiber orientations of each lamina.

In this paper a laminate made by graphite epoxy lamina of constant thickness t was considered, the elastic properties and thickness of the material are the following:

$$E_{11} = 127.6 \text{ GPa}; E_{22} = 13.0 \text{ GPa}; G_{12} = 6.4 \text{ GPa}; \nu_{12} = 0.3.$$

The ply thickness is $t = 0.127 \text{ mm}$.

The laminate has length $a = 0.508 \text{ m}$, width $b = 0.254 \text{ m}$, and is made by 64 plies. total thickness $t = 8.128 \text{ mm}$, [9]. The only design variables are hence the angles θ_k of each lamina. We applied ACOR with Gaussian perturbation and ACOR_L with Lévy perturbation to a different set of allowed fiber orientations and of different constraints on the laminate stacking sequence able to reduce the number of independent variables. Table 3 shows the different set of possible fiber orientations, the constraint adopted on the stacking sequence and the number of independent variables for each case analyzed in the present paper.

The *continuous relaxation approach* is adopted in the optimization algorithm, i.e. the discrete variables are replaced by continuous ones and in the evaluation of the objective function are transformed in the allowed discrete values. This choice is suitable due to the natural order in the design variables space.

Table 3. Design problems analyzed

| Fiber directions | Constraints | No. design variables |
|-------------------------------------|---------------------|----------------------|
| $P_1 = [0, 45, 90]$ | symmetric, balanced | 16 |
| $P_2 = [0, 30, 60, 90]$ | symmetric | 32 |
| $P_3 = [0, 15, 30, 45, 60, 75, 90]$ | symmetric | 32 |

The ACOR parameters values has been taken the same adopted for the function optimization problems, for α the following values have been taken {2.0, 1.9, 1.8}. The maximum number of function evaluations has been set to 5000. Table 4 shows the results obtained with different values of α . Also in this case the behaviour is similar to that found for the mathematical functions with ACOR_L performances better than classical ACOR ones.

Table 4. Mean of the best solution over 100 independent runs, from standard ACOR and ACOR_L. The number in the parentheses indicate standard deviations. The last row shows the t-test between $\alpha=2.0$ and $\alpha=1.8$ results. The asterisk indicates that the difference is not negligible.

| | P ₁ | P ₂ | P ₃ |
|-------------------------------|----------------|----------------|----------------|
| ACOR _L | 3972.48 | 4076.84 | 4103.19 |
| ($\alpha=1.8$) | (0.73) | (7.50) | (6.63) |
| ACOR _L | 3972.32 | 4074.57 | 4100.40 |
| ($\alpha=1.9$) | (0.98) | (13.90) | (9.85) |
| ACOR | 3972.31 | 4072.76 | 4097.67 |
| ($\alpha=2.0$) | (1.02) | (13.34) | (17.04) |
| t-test | | | |
| $\alpha=2.0$ vs. $\alpha=1.8$ | 1.33 | 2.67* | 3.02* |

5 Conclusions

In this paper, a new perturbation operator, based on Lévy distribution, is proposed for Ant Colony Optimization in continuous domains. The modified algorithm is here called ACOR_L. The behaviour of the algorithm in some interesting real world problems has been observed and studied in the paper. In particular the ACOR_L has been applied to a difficult multi-modal problem of composite laminates buckling load maximization. As it can be noted the wider exploration potential of the Lévy distribution allows the algorithm in case of multimodal functions showing many local minima to attain statistically significant better performance than standard ACOR. Further studies will be addressed towards the implementation of an adaptive version of ACOR_L.

References

- Herrera, F., Lozano, M., Verdegay, J.L.: Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. Artificial Intelligence Review 12, 265–319 (1998)

2. Back, T.: Evolution Strategies: an alternative evolutionary algorithm. *Artificial Evolution* 1063, 3–20 (1995)
3. Dorigo, M.: Optimization, learning and natural algorithms (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy (1992)
4. Dorigo, M., Gambardella, L.M.: Ant colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evol. Comp.* 1(1), 53–66 (1997)
5. Socha, K., Dorigo, M.: Ant colony optimization for continuous domains. *European Journal of Operational Research* 185, 1155–1173 (2008)
6. Lee, C., Yao, X.: Evolutionary programming using mutations based on Lévy probability distribution. *IEEE Trans. on Evol. Comp.* 8(1), 1–13 (2004)
7. Gutowski, M.: Lévy flights as an underlying mechanism for global optimization algorithms. *Math-ph/0106003* (2001), <http://arxiv.org/pdf/math-ph/0106003>
8. Gurdal, Z., Haftka, R.T.: Optimization of composite laminates. In: NATO Advanced Study Institute on Optimization of Large Structural Systems, Germany (1991)
9. Soremekun, G., Gurdal, Z., Haftka, R.T., Watson, L.T.: Composite laminate design optimization by genetic algorithm with generalized elitist selection *Computers and Structures* 79, 131–143 (2001)
10. Erdal, O., Sonmez, F.O.: Optimum design of composite laminates for maximum buckling load capacity using simulated annealing *Composite Structures* 71, 45–52 (2005)
11. Karakaya, S., Soyksap, O.: Buckling optimization of laminated composite plates using genetic algorithm and generalized pattern search algorithm. *Struct. Multidisc. Optim.* 39, 477–486 (2009)
12. Lévy, P.: Théorie des erreurs la loi de Gauss et les lois exceptionnelles. *Bulletin de la Société Mathématique de France* 52, 49–85 (1924)
13. Samorodnitsky, G., Taqqu, M.S.: Stable non-Gaussian random processes: Stochastic models with infinite variance. Chapman and Hall, New York (1994)
14. Grigoriu, M.: *Stochastic Calculus Applications in Science and Engineering*. Birkhäuser, Boston (2002)
15. Reddy, J.N.: *Mechanics of Laminated Composite Plates and Shells*, 2nd edn. CRC press, Boca Raton (2004)

Teaching Assignment Problem Solver

Ali Hmer and Malek Mouhoub

University of Regina
Wascana Parkway, Regina, SK, Canada, S4S 0A2
{hmer200a,mouhoubm}@cs.uregina.ca

Abstract. In this paper, we describe an extension approach to the backtracking with look-ahead forward checking method that adopts weighted partial satisfaction of soft constraints that has been implemented to the development of an automated teaching assignment timetabling system. Determining the optimal solution for a teaching assignment problem is a challenging task. The objective is to construct a timetable for professors from already scheduled courses that satisfy both hard constraints (problem requirements such as no teacher should be assigned two courses at the same time) and soft constraints (teacher preferences) based on fairness principle in distributing courses among professors. The approach is done mainly to modify the variable selection method and the value assignment technique taking into account preferences and based on fairness principle. The optimized look-ahead backtracking method applied to the solution is presented and discussed along with computational results.

Keywords: Teaching Assignment Problem, Soft Constraints, Constraint Optimization.

1 Introduction

In this paper, we describe a constraint programming system, with a web site as front-end to demonstrate a suggested solution technique of the Teaching Assignment Problem. The timetabling problem in general is mostly, if not always, an over constrained combinatorial optimization problem and hence it is considered one of the most difficult problems to solve. The Teaching Assignment Problem in essence is a branch of the timetabling problem and it is the problem of assigning professors to time slots that is occupied by courses in a specific week. The resulted weekly timetable is to be used to organize the teaching process at a university or any educational institute given that the courses have already been scheduled over the time slots and rooms. Each professor is assigned a total number of courses to be taught that should not be violated. Each professor is allowed to express interest or dislike in certain courses through weighed preferences that can or cannot be satisfied. Some professors can be assigned some courses in advance. The final solution should be constructed based on distributing the given courses over professors based on fairness principle as well as maximizing the total weight of the solution. The total weight of a solution is the sum of all satisfied preferences. The literature is very rich on the topic of university

timetabling in general as there are different ways to solve the problem; most of them depend on specific needs considered by the institution that the timetabling is designed for. However, to the knowledge of the authors, no literature is dealing with the teaching assignment problem. In our case, we considered the problem as two-fold stages. The first is to assign courses to rooms and time slots and the second is to assign professors to the resulting time slots with courses. In this study, we only tackled the second one. Timetabling problems, in general, are usually over constrained as it is not always possible to satisfy all requirements. User preferences can be used to relax these requirements. In our study case, we have used a more specific model with preferences which utilizes weight for each constraint and try to maximize the total weight of satisfied soft constraints. As a development approach, our work includes a development of a solver for soft constraints. The solver was implemented by the authors as an extension of a well-known CSP solver named "Java Cream" [1] to include soft constraints in the backtracking mechanism which the Java Cream Solver is lacking. The solver itself was re-coded entirely, by the author, using Microsoft C# language from Java language. Some of the optimized technologies introduced in C# and in .NET framework, such as LINQ, were used to enhance and optimize the local search.

The next section of this paper provides a related work for the problem. Section 3 provides a description to the teaching assignment problem. The added soft-constraint approach that was implemented within the solver along with the modified search algorithm developed for this problem is detailed in section 4. This includes a description of how the problem has been solved as well as the representation of soft and hard constraints. Furthermore, a discussion on how the search is done is provided at the end of this section. Section 5 provides a description for the web based system used to implement the solver. Computational results are discussed in Section 6. The final section reviews the results of our work and looks to future extensions of the problem solution and soft-constraint solver improvements.

2 Related Work

Over the last 30 years, the timetabling problem is considered to be one of the broadly studied scheduling problems in Artificial Intelligence and Operations Research literature [2]. Educational timetabling, to be specific, has been the main topic of quite few papers in various scientific journals and the topic of many theses in academia society. The course timetabling problem deals effectively with courses and time slots which have to be scheduled during the academic term. The problem basically is the scheduling of a known number of courses into a known number of time slots spread all over the week in such a way that constraints are satisfied.

As there are many versions of the Timetabling Problem, a variety of techniques have been used to solve it [3], [4]. Most of these techniques range from graph colouring to heuristic algorithms. Another focus of research in the timetabling

problem was on the application of a single solution approach which in effect a large variety of such approaches have been tried out, such as an integer programming approach [4], Tabu search [3], and Simulated Annealing [5]. Recently, some researchers have attempted to combine several approaches, such as hybridization of exact algorithms and Meta-heuristics. One of the most primitive methods used to solve this problem is graph colouring in which vertices represent events where two vertices are connected if and only if there is a conflict. [5], [6], [7], [8] and [9] proposed a number of formulations by graph colouring for a set of class teacher timetabling problems and discussed the inherent complexity. In [10], graph colouring has been used to solve course and exam timetabling. Linear programming models were also used to formulate the course time-tabling problem usually with binary variables [11], [12], [13], and [14]. An Integer Programming approach [15] was also used to model the timetabling problem as assignment problem with numerous types of constraints and large number of binary or integer variables. Rudov and Murray introduced an extension of constraint logic programming [16] that allows for weighted partial satisfaction of soft constraints is implemented to the development of an automated timetabling system. In [17], an Evolution Strategy to generate the optimal or near optimal schedule of classes is used to determine the best, or near best timetable of lecture/courses for a university department. Case Based Reasoning is another approach that has recently been applied to university timetabling [18], [19], [20], and [21]. Case Based Reasoning is believed to be studied as early as 1977 with the study of Schank and Abelson [22]. Case Based Reasoning has also been successfully applied to scheduling and optimization problems. Burke et al. [23] also, in a published article, developed a graph-based hyper-heuristic (GHH) which has its own search space that operates in high level with the solution space of the problem generated by the so-called low level heuristics.

3 Problem Description

In general, the timetabling problem is the assignment of time slots to a set of events. These assignments usually include many considerable constraints of different types. At the department of Computer Science, University of Regina, in any term, the timetabling process currently consists of constructing a class schedule prior to student registration. The professors and classes timetabling problem [24] and [25] is NP-Complete. The teaching assignment problem is the problem of assigning courses, scattered over time slots, to professors. In our case, the teaching assignment problem is described as follows.

1. There is a finite set of courses $C = \{c_1, c_2, \dots, c_{|C|}\}$ and a finite set of time slots $T = \{t_1, t_2, \dots, t_{|T|}\}$, which already have been assigned to courses C . This is typically provided as courses occupy time slots. So each course could occupy just one time slot, usually 3 hours; two time slots, usually an hour and half each; or 3 time slots, usually an hour each. For any course, the time slots assigned to it must not overlap. t_i can be assigned to different

courses as long as they are in different rooms and different professors. Our approach is nothing to do with these assignments as these assignments are considered as input for the problem to solve.

2. There is a finite set of professors $P = \{p_1, p_2, \dots, p_{|P|}\}$.
3. In this scheduling problem, courses represent variables while professors represent variables domain values.
4. The problem is to schedule P to C in a way such that no professor p_i is in more than one place at a time t_i .
5. The constraints for this problem are soft and hard. The soft constraints should not all be satisfied and on the contrary all hard constraints must be satisfied so a possible solution to the problem is one that satisfies all the hard constraints but not necessary soft constraints.
6. Soft Constraints are preferences that do not deal with time conflicts and have weight (or Cost) associated with them. Our goal is to maximize the total weight of a solution (or minimize the total cost). We have two types of soft constraints; the first is count, where a professor has a maximum number of courses assigned to him that should not be exceeded. The second is preferences that any professor can express as interest or dislike in certain courses which have weights (or costs). This type of constraints can or cannot be satisfied.
7. In our case, we have two types of soft constraints, both of them related to professors preferences. These preferences named equal and not equal, which is indicate if a professors provided an interest or dislike in a that course (variable).
8. Hard Constraints are typically constraints that physically cannot be violated. This includes time slots that must not overlap in time, and in our problem time slots that overlap in time must not be taught by the same professor. There is another type of hard constraints where time slots represent a course can be assigned to a professor in advance prior to starting the search for a solution.
9. There is a total weight function that measures the quality of the current solution. The object of this function is to return the sum of all weights/costs associated with the satisfied preferences. The aim of the optimization technique is to maximize the total weight function or minimize the total cost.

4 Algorithm Description

As mentioned above, the solver, used in solving the problem, is re-coded from a well-known solver named "*Java Cream*" using Microsoft C# language. The original solver can be used to model any constraint satisfaction or optimization on finite domains problems [26]. However, it lacks any proper handling of soft constraints. As known, any timetabling/scheduling problem would be mostly over constrained and therefore it cannot be solved unless constraints are relaxed. Hence, the necessity came to add soft constraints as part of the re-coded solver to solve timetabling problems. The backtracking method is the one that was

modified to take into account soft constraints. Although we describe only the modified backtracking method, which is used by two of the five methods that the solver adopts: Branch and Bound; and Iterative Branch and Bound. However, because preliminary tests showed that performance is not significantly improved in our application when using the other three methods; Taboo Search, Random Walk and Simulated Annealing, we only consider the backtracking method to meet our requirements. Furthermore, all solver search methods use the same variable/value ordering methods and hence would use the same approach mentioned here. We think that because the application study case variables and values are relatively small, the tests performance using other methods has not improved but might be better if another application is implemented which might have more complex variables and values.

Basically, the backtracking algorithm [27] has two phases. The first stage is what is called "a forward phase" in which the variables are selected sequentially and the current partial solution is extended by assigning a consistent value for the next variable if one exists. The second phase is known as "a backward phase" in which the algorithm returns to the previous assigned variable when no consistent solution exists for the current variable. For the forward phase, the adopted solver originally decides which variable to instantiate next by selecting the one that has minimum number of domain values (i.e. the one that its domain size is minimum). Then the solver decides which value to assign to the next variable by assigning the maximum value in the variable domain. By assigning a value to a variable, this value is eliminated from all other variables' domains. This is known as look-ahead backtracking.

The variables in the original solver are ordered according to their domain size and the variable with the highest domain size is first. In the modified solver, you still can use the same mechanism, but when "soft constraints approach" is used, variables are ordered by the highest weight on soft constraints and then on highest domain size.

The values in the original solver are ordered incrementally. However, in the modified solver that uses soft constraints approach, values are ordered by values associated to equal soft constraints that least have been assigned to any variable before first and then other values incrementally and last values that are associated with not equal soft constraints.

Because of the soft constraints that have been added to the solver, we have improved the two backtracking phases as follows if "soft constraint approach" method is selected in solving the problem:

1. On deciding which variable (Course) to instantiate next, the solver tries primarily to select the variable with the highest weight on equal soft constraints that have not been assigned a value (Its domain size is greater than one); if not then it will return randomly one of the variables (courses). The idea behind this is to try to select a variable that has soft constraints associated with it first, if not found then it will act on the other types of variables.
2. If the previous hint is not implemented, it will give the chance to assign values to variables that do not have soft constraints with them where they should

have been at least trying to be assigned to variables with soft constraints. In this case, variables with preferences will miss the chance to get their preferences assigned to them.

3. On deciding which value to assign to the variable selected in the previous step, a method, first, checks if there are equal soft constraints associated to that variable. If there are not any, then it will randomly select a value from its domain (i.e. domain values represent professors).
4. It is worth mentioning that even if there are no soft constraints associated with it, the method tries to not to choose a value that is associated with another variable that has an equal soft constraints as it might be needed in a later stage.
5. If there are indeed equal soft constraints associated to that variable, then it assigns the value that least has been assigned to any variable before. This is in compliance with the "fairness" principle. Furthermore, the value is selected randomly if there is more than one value.

5 Web Based Interface for the Teaching Assignment Problem Solver System (TAPS)

We have implemented a web-based application for solving the timetabling problem. The idea behind this approach is to get professors to enter their preferences through the web site. Web based applications generally are more convenient for users. For instance, every professor can enter his/her preferences from office/home and there is no need to provide this information to application operator to enter their data. The Teaching Assignment Problem Solver (TAPS) was

Main Menu

- Home
- Course
- Professor
- Solution
- Settings
- About this

Links

- C-Sharp Cream Solver
- Java Cream Solver
- ASP.NET MVC
- NHibernate

Professors
Create - Edit - Delete Professors

[Add New Professor](#)

| Title & Name | # of Courses | Preferences | Edit | Delete |
|----------------------|--------------|-------------|------|--------|
| Dr. David Newton | 1 | 0 wiegh 0 | Edit | Delete |
| Dr. Dion Griffiths | 2 | 1 wiegh 5 | Edit | Delete |
| Dr. George Malbourne | 1 | 2 wiegh 9 | Edit | Delete |
| Dr. Ian Morrison | 2 | 0 wiegh 0 | Edit | Delete |
| Dr. John Smith | 3 | 3 wiegh 12 | Edit | Delete |
| Dr. Lee Yang | 2 | 1 wiegh 5 | Edit | Delete |
| Dr. Linda Parkinson | 2 | 1 wiegh 5 | Edit | Delete |
| Dr. Peter Murphy | 2 | 0 wiegh 0 | Edit | Delete |
| Dr. Philip Stanley | 1 | 2 wiegh 9 | Edit | Delete |
| Dr. Scott Howard | 2 | 2 wiegh 9 | Edit | Delete |

Fig. 1. Professors information page

Professors
Create - Delete Preferences for [Dr. John Smith]

[Add New Preference \[Max 5\]](#)

| Course Name ▲ | Weight ◆ | Type ◆ | |
|---------------|----------|----------------|--------|
| CS110-001 | 3 | Interested | Delete |
| CS301 | 5 | Interested | Delete |
| CS350 | 4 | Not Interested | Delete |

[Go back to Professors](#)

Fig. 2. Professors' preferences information page

Solutions
[Generate another set of solutions](#)

100 Solution(s) found in 0.4135 second(s)

Viewing Solution No. 1 Solution Weight: 40
 Time spent to generate 100 solution(s): 413.5 ms (0.4135 seconds)
 Time spent to generate this solution: 0.6 ms (0.0006 seconds)

[First Solution](#)
 [Prev. Solution](#)
 [Next Solution](#)
 [Last Solution](#)

| Courses | | | Professors | | |
|---------|-------------|----------------------|------------|----------------------|----------------------|
| ▲ | Course Code | ◆ Professor Name | ▲ | Professor Name | ◆ # Assigned Courses |
| 1 | CS110-001 | Dr. John Smith | 1 | Dr. David Newton | 1 |
| 2 | CS110-002 | Dr. Goerge Malbourne | 2 | Dr. Dion Griffiths | 2 |
| 3 | CS115 | Dr. Ian Morrison | 3 | Dr. Goerge Malbourne | 1 |
| 4 | CS201 | Dr. Peter Murphy | 4 | Dr. Ian Morrison | 2 |
| 5 | CS210 | Dr. Peter Murphy | 5 | Dr. John Smith | 3 |
| 6 | CS215 | Dr. David Newton | 6 | Dr. Lee Yang | 2 |
| 7 | CS261 | Dr. Lee Yang | 7 | Dr. Linda Parkinson | 2 |
| 8 | CS301 | Dr. Scott Howard | 8 | Dr. Peter Murphy | 2 |
| 9 | CS305 | Dr. Philip Stanley | 9 | Dr. Philip Stanley | 1 |
| 10 | CS320 | Dr. Linda Parkinson | 10 | Dr. Scott Howard | 2 |
| 11 | CS325 | Dr. Linda Parkinson | | | |
| 12 | CS330 | Dr. Scott Howard | | | |
| 13 | CS340 | Dr. John Smith | | | |
| 14 | CS350 | Dr. Ian Morrison | | | |

Fig. 3. Solution page

developed using Microsoft ASP.NET MVC (Model-View-Controller) as an interface, Microsoft SQL Server 2005 as database engine and Internet Information Server (IIS) as web server. The MVC model provides a rich graphic user interface using HTML and JQuery (Java script based library). There are four main parts for the web site. The first is devoted to courses management and its information can be entered by administrator. The information includes the course assigned week days, assigned time slots, and assigned professor (if needed). The later will

be treated by the solver as hard constraint. The interface also displays the number of professors interested and not interested in that course. The whole courses are displayed as a table where there is an option for inserting, editing and deleting a course. The second is dedicated to professors' information management and their information can be entered by professors themselves. The information includes the number of courses that can be assigned to each professor (i.e. count constraint) and the professors' preferences. Both preferences will be dealt by the solver as soft constraints. The third is for searching for a solution using the information provided and using the C# cream as a background solver. The solution section provides an interface for searching for solutions using the information provided in the previous two sections. If solutions are found, they will be displayed in this section's web page. Among the information displayed, there is time spent for generating all solutions and the time spent for each solution along with the solution weight. There is also the option to display the next and previous solution. There are three tables; the first one is the main table where courses are displayed with professors. The second table displays professor's names and the number of assigned courses. The displays all constraints (hard and soft) used in finding the solutions. The last section is for Web site settings. This includes the number of hours per course, maximum break minutes per session, maximum number of courses per professor, number of preferences per professors, maximum number of generated solutions, the option to generate only better solutions in terms of solution weight and the maximum timeout that should be used in the solver to generate solutions. The screen shots below illustrate the professors and solutions sections.

6 Experimental Tests and Results

The experimental tests compare between our proposed approach and the original backtracking method.

In order to do tests on the designed Web site and the proposed solver, we used data from the Computer Science department at the University of Regina for both courses and professors information including courses time slots. Then we assigned randomly some of the courses to some professors and assigned professors to show some interest and dislikes in some of the courses.

Overall, we used 17 courses as solver variables and 10 professors as solver values. From these courses we entered interest in 4 courses for different professors and disinterest in just one course. Experimental computations were done with a number of objectives in mind. The main goal was to provide a table of courses assigned to professors where all hard constraints are satisfied and the weight of soft constraints is maximized.

The second experiment involved using the same solver to solve a problem with the same variables and domain values but using non preference approach (The original backtracking method).

We have also set the solver to generate the first 100 solutions considering the first solution is the most optimized one and to generate only same or better weighted

solutions and the solver has only 100 seconds to generate any solution at any given time. We used a PC with the following capabilities: Core 2 Duo Quad processor (2.4 GHz) with 6 GB ram. We have asked the solver to search for solutions 10 times to get a bigger picture of the search time spent in finding solutions.

The results showed that the average time to find a solution was between 1.92 ms and 2.567 ms. When using non-preference approach (i.e. original solver's ordinary variable/value ordering), we were unable to find a feasible solution that can satisfy the maximum number of soft constraint in the first 100 solutions. On the contrary, when using the preference approach, the first 10 solutions were optimal for our problem that satisfied hard and soft constraints.

7 Conclusion

We have successfully applied modified back tracking method to solve the teaching assignment problem. Feasible schedules were obtained for real data sets, including professors' preferences without the need for a huge computational effort. The original solver was meant to solve integer based variables problems but for problems with hard constraints. We have extended the solver to adopt soft constraints and we think that it has been a success. In conclusion, this application of Teaching Assignment Problem Solver appears to be quite successful and we are satisfied and ready to implement it to generate actual schedules for future terms. We also think that this approach can be implemented in similar problems like Exam Supervision scheduling. Based on the gathered experience of this test, we concluded that this approach is computationally feasible.

References

1. Tamura, N.: Cream: Class Library for Constraint Programming in Java. Kobe University (2009), <http://bach.istc.kobe-u.ac.jp/cream/>
2. Valdes, R.A., Crespo, E., Tamarit, J.M.: Design and implementation of a course scheduling system using Tabu Search. *European Journal of Operational Research* 37, 512–523 (2002)
3. Burke, E.K., Jackson, K., Kingston, J., Weare, R.E.: Automated university timetabling: the state of the art. *The computer journal* 40, 565–571 (1997)
4. Carter, M.W., Laporte, G.: Recent developments in practical course timetabling. In: Burke, E.K., Carter, M. (eds.) PATAT 1997. LNCS, vol. 1408, pp. 3–19. Springer, Heidelberg (1998)
5. Welsh, D.J.A., Powell, M.B.: An upper bound for the chromatic number of graph and its application to timetabling problems.1. *The Computer Journal* 10, 360–364 (1967)
6. Wood, D.C.A.: Technique for colouring a graph applicable to large scale timetabling problems. *The Computer Journal* 12, 317–319 (1969)
7. Selim, S.M.: Split Vertices in Vertex colouring and their application in developing a solution to the faculty timetable problem. *The Computer Journal* 31, 76–82 (1988)
8. Burke, E.K., Ross, P. (eds.): PATAT 1995. LNCS, vol. 1153, pp. 296–308. Springer, Heidelberg (1996)
9. Miner, S., Elmohamed, S., Yau, H.W.: Optimizing Timetabling Solutions Using Graph Coloring. NY: NPAC REU program, NPAC, Syracuse University (1995)

10. Timothy, A.R.: A Study of university timetabling that blends graph coloring with the satisfaction of various essential and preferential conditions. Rice University: Ph.D. Thesis (2004)
11. Daskalaki, S., Birbas, T., Housos, E.: An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 117–135 (2004)
12. Daskalaki, S., Birbas, T.: Efficient solutions for university timetabling problem through integer programming. *European Journal of Operational Research*, 106–121 (2005)
13. Dimopoulou, M., Miliotis: An Automated Course Timetabling System developed in a distributed Environment: a Case Study. *European Journal of Operational Research*, 153, 136–148 (2004)
14. Dimopoulou, M., Miliotis, P.: Implementation of a University Course and Examination Timetabling System. *European Journal of Operational Research* 130, 202–213 (2001)
15. Schimmelpfeng, k., Helber, S.: Application of a real-world university-course timetabling model solved by integer programming. Springer, Heidelberg (2006)
16. Rudov, H., Murray, K.: University Course Timetabling with Soft Constraints, pp. 310–327. Springer, Heidelberg (2003)
17. George, T.B., Opalikhin, V., Chung, C.J.: Using an Evolution Strategy for a University Timetabling System with a Web Based Interface to Gather Real Student Data. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O’Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724. Springer, Heidelberg (2003)
18. Burke, E.K., MacCathy, B., Petrovic, S., Qu, R.: Case-based reasoning in course timetabling: an attribute graph approach. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 90–105. Springer, Heidelberg (2001)
19. Burke, E.K., MacCathy, B., Petrovic, S., Qu, R.: Multiple-retrieval case-based reasoning for course timetabling problems. *Journal of the Operational Research Society*, 1–15 (2005)
20. Burke, E.K., MacCathy, B., Petrovic, S.: Knowledge discovery in a hyperheuristic for course timetabling using case-based reasoning. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 90–103. Springer, Heidelberg (2003)
21. Burke, E.K., MacCathy, B., Petrovic, S., Qu, R.: Structured case in case-based reasoning-re-using and adapting cases for timetabling problems. *Knowledge-Based Systems* 13, 159–165 (2000)
22. Schank, R.C., Abelson, R.P.: Scripts, plans, goals and understanding. Erlbaum, New Jersey (1977)
23. Burke, E.K., McCollum, B., Meisels, A., Petrovic, S., Qu, R.: A graph-based hyperheuristic for educational timetabling problem. *European Journal of Operational Research*, 1–16 (2006)
24. Gislen, L., Soderberg, B., Peterson, C.: Teachers and Classes with Neural Nets. *International Journal of Neural Systems* 1, 167–168 (1989)
25. Gislen, L., Soderberg, B., Peterson, C.: Complex scheduling with Potts neural networks. *Neural Computation* 4, 805–831 (1992)
26. Tamura, N.: Calc/Cream: OpenOffice Spreadsheet Front-End for Constraint Programming. In: Umeda, M., Wolf, A., Bartenstein, O., Geske, U., Seipel, D., Takata, O. (eds.) INAP 2005. LNCS, vol. 4369, pp. 81–87. Springer, Heidelberg (2006)
27. Dechter, R.: Constraint Processing, 1st edn., pp. 123–128. Morgan Kaufmann, San Francisco (2003)

Swarm Control Designs Applied to a Micro-Electro-Mechanical Gyroscope System (MEMS)

Fábio Roberto Chavarette^{1,3}, José Manoel Balthazar², Ivan Rizzo Guilherme²,
and Orlando Saraiva do Nascimento Junior³

¹ Faculty of Engineering, UNESP – Univ Estadual Paulista, DM, Avenida Brasil,
56, 15385-000, Ilha Solteira, SP, Brazil

² Geoscience and Exact Science Institute, UNESP – Univ Estadual Paulista, DEMAC,
PO BOX 178, 13500-230, Rio Claro, SP, Brazil

³ Ometto Herminio University Center at Araras – UNIARARAS, Engineering Center, Av.
Dr. Maximiliano Baruto, 500, Jd. Universitário, 13607-339, Araras, SP, Brazil
chavarette@gmail.com, {jmbaltha, ivan}@rc.unesp.br,
saraiva@uniararas.br

Abstract. This paper analyzes the non-linear dynamics of a MEMS Gyroscope system, modeled with a proof mass constrained to move in a plane with two resonant modes, which are nominally orthogonal. The two modes are ideally coupled only by the rotation of the gyro about the plane's normal vector. We demonstrated that this model has an unstable behavior. Control problems consist of attempts to stabilize a system to an equilibrium point, a periodic orbit, or more general, about a given reference trajectory. We also developed a particle swarm optimization technique for reducing the oscillatory movement of the nonlinear system to a periodic orbit.

Keywords: Particle Swarm Optimization, MEMS Gyroscope, Evolutionary Algorithms.

1 Introduction

The field of micro machining is forcing a profound redefinition of the nature and attributes of electronic devices. The technology of micro electro mechanical systems (MEMS) has found numerous applications in recent years, for example the electromechanically filters, biological and chemical sensing, force sensing and scanning probe microscopes [1-4].

This technology allows motion to be incorporated into the function of micro scale devices. However, the design of such mechanical systems may be quite challenging due to nonlinear effects that may strongly affect the dynamics.

Microscopic gyroscopes [5, 6] are helping enable an emerging technology called electronic stability control. The resulting system helps prevent accidents by automatically activating brakes on out-of-control vehicles. The technology may be particularly useful for vehicles with a higher center of gravity, which makes them prone to rolling.

Electronic stability control is available in luxury vehicles, but sensors made from quartz were too expensive for widespread installation. Innovations in MEMS gyroscope technology make these systems more affordable.

Control problems consist of attempts to stabilize an unstable system to equilibrium point, a periodic orbit, or more general, about a given reference trajectory. In the last years, a significant interest in control of the nonlinear systems, exhibiting unstable behavior, has been observed and many of the techniques discussed in the literature [7-9]. Among strategies of control with feedback the most popular is OGY (Ott-Grebogi-York) method [7]. This method uses the Poincaré map of the system. Recently, a methodology, based on the application of the Lyapunov-Floquet transformation, was proposed by Sinha et al. [8] in order to solve this kind of problem. This method allows directing the chaotic motion to any desired periodic orbit or to a fixed point. It is based on linearization of the equations, which described the error between the actual and desired trajectories. Another one technique was proposed by Rafikov and Balthazar in [9], where the Dynamic Programming was used to solve the formulated optimal control problems. Several techniques that can be applied to a wide range of problems.

Different from numerical approach, mentioned above, there are other algorithm based approach. In this sense, here we proposed the algorithm based approach, that used particle swarm optimization (PSO) algorithms. The PSO algorithms is a population based stochastic optimization technique developed by Kennedy and Eberhart in 1995 [10]. Using PSO algorithms allows directing the chaotic motion to any desired periodic orbit or to a fixed point. In this work, we proposed and develop a PSO based optimization algorithms for control the unstable movement of MEMS gyroscope.

The paper is outlined as follows. In Section are showed the concepts related with the nonlinear model to the MEMS gyroscope. In Section 3, is shown the application of the Particle Swarm Optimization algorithm in the MEM gyroscope. In Section 4, the proposed control swarm approach are presented. In Section 5, we do some concluding remarks of this work. In section 6, we list the main bibliographic references used.

2 MEMS Gyroscope Model

The technology of micro electro mechanical systems (MEMS) has found numerous applications in recent years, for example, the MEMS gyroscope (Fig. 1).

Here, we consider a mechanical model and the derivation of governing equations done by [5] for the MEMS gyroscope, commonly function on the coupling of two linear resonant modes via the Coriolis force.

The micro gyroscope device consists itself of a perforated proof mass constrained to move in the plane by a suspension of micro beams. It is forced along one axis, the so-called drive axis, by a set of non-interdigitated comb drives, and its motion along the other axis, is detected by a set of parallel plate capacitors.

The governing equations of motion of MEMS gyroscope were obtained by [5] and they are:

$$\begin{aligned} m\ddot{x} + c\dot{x} + [k_{11} + r_1V_a^2(1 + \cos(2wt))]x + [k_{31} + r_3V_a^2(1 + \cos(2wt))]x^3 - 2\Omega\dot{y} &= 0 \\ m\ddot{y} + c\dot{y} + k_{12}y + k_{32}y^3 + 2\Omega\dot{x} &= 0 \end{aligned} \quad (1)$$

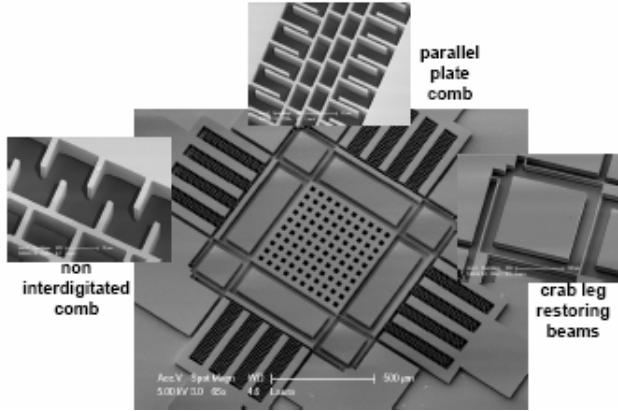


Fig. 1. Micrograph of the micro gyroscope [5]

And by using the non-dimensionless variables

$$\left. \begin{aligned} q_d &= \frac{x}{L}, \quad q_s = \frac{y}{L}, \quad \tau = t \sqrt{\frac{m}{k_{11}}}, \quad \varepsilon \zeta = \frac{c}{\sqrt{mk_{11}}}, \quad \varepsilon \lambda_1 = \frac{r_1 V_a^2}{k_{11}}, \quad \varepsilon \nu_3 = \frac{k_{31} L^2}{k_{11}} \\ \varepsilon \lambda_3 &= \frac{r_3 L^2 V_a^2}{k_{11}}, \quad \varepsilon \gamma = 2\Omega \sqrt{\frac{m}{k_{11}}}, \quad \varepsilon \delta = \frac{k_{12}}{k_{11}} - 1, \quad \varepsilon \xi = \frac{k_{32} L^2}{k_{11}} \end{aligned} \right\} \quad (2)$$

We will obtain:

$$\begin{aligned} q_d'' + 2\varepsilon \zeta q_d' + [1 + \varepsilon \lambda_1 (1 + \cos(2\omega t))] q_d + \\ [\varepsilon \nu_3 + \varepsilon \lambda_3 (1 + \cos(2\omega t))] q_d^3 - \varepsilon \gamma q_s' = 0 \\ q_s'' + 2\varepsilon \zeta q_s' + (1 + \varepsilon \delta) q_s + \varepsilon \xi q_s^3 + \varepsilon \gamma q_d' = 0 \end{aligned} \quad (3)$$

By applying the method of averaging in (3) and rewriting the equations of the dynamical system, in state form, the governing equations may be written as being[5]:

$$\begin{aligned} \dot{x}_1 &= \frac{\varepsilon}{8} [4x_2 \gamma \cos(x_3 - x_4) + x_1 (-8\zeta + (2\lambda_1 + \lambda_3 x_1^2) \sin(2x_1))], \\ \dot{x}_2 &= -\frac{\varepsilon}{2} [x_1 \gamma \cos(x_3 - x_4) + 2\zeta x_2] \\ \dot{x}_3 &= \frac{\varepsilon}{8x_1} [-4x_2 \gamma \sin(x_3 - x_4) + x_1 (4\lambda_1 - 8\sigma + 3(\nu_3 + \lambda_3) x_1^2 + 2(\lambda_1 + \lambda_3 x_1^2) \cos(2x_3))], \\ \dot{x}_4 &= \frac{\varepsilon}{8x_2} [-4x_1 \gamma \sin(x_3 - x_4) + x_2 (4\delta - 8\sigma + 3\xi x_2^2)] \end{aligned} \quad (4)$$

Here the parameter x_1 is the amplitude of oscillation the drive axis, and x_2 is the amplitude of oscillation along the sensing axis. The variables x_3 and x_4 are the phases of oscillation for the two axes.

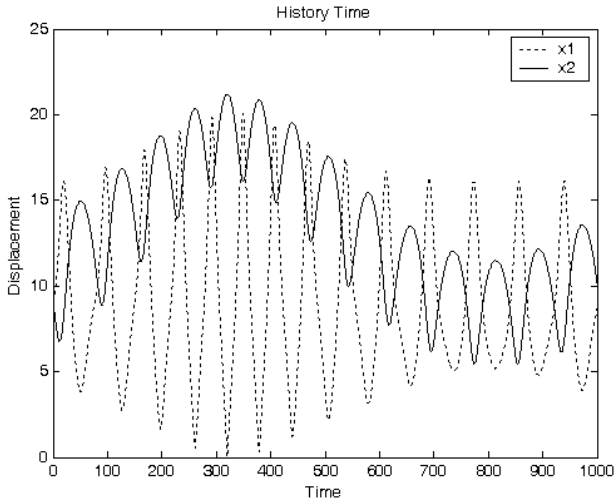


Fig. 2. Dynamical behavior of the time history: x_1 and x_2

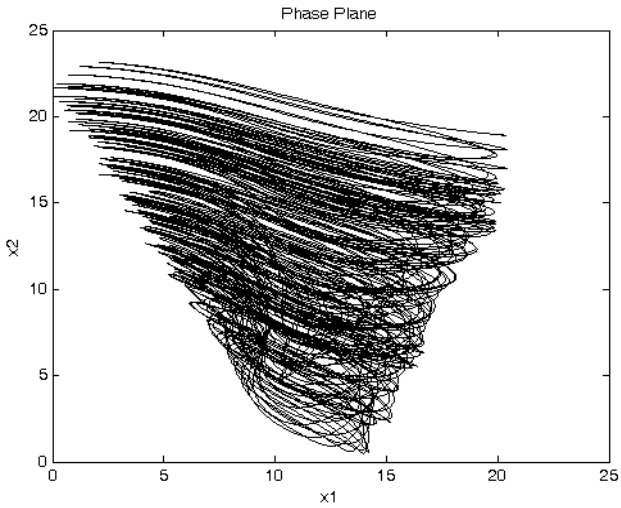


Fig. 3. Phase Portrait

In the Figure 2-5 is showed the dynamics behavior of the adopted dynamics model, by using numerical values, for the chosen parameters $\epsilon=0.001$; $\lambda_1=1$; $\lambda_3=2$; $\zeta=0.1$; $\gamma=56$; $\xi=0.1$; $v_3=0.01$; $\delta=-0.01$; $\sigma=\delta/2$; $x_1=9$; $x_2=9$; $x_3=9$ and $x_4=9$.

In the Figure 2 is showed the dynamics behavior of time history for the x_1 and x_2 .

In the Figure 3 shown the phase portrait for x_1 and x_2 .

In the Figure 4 shows the diagram of the stability for x_1 (with the region's control applied it is illustrate).

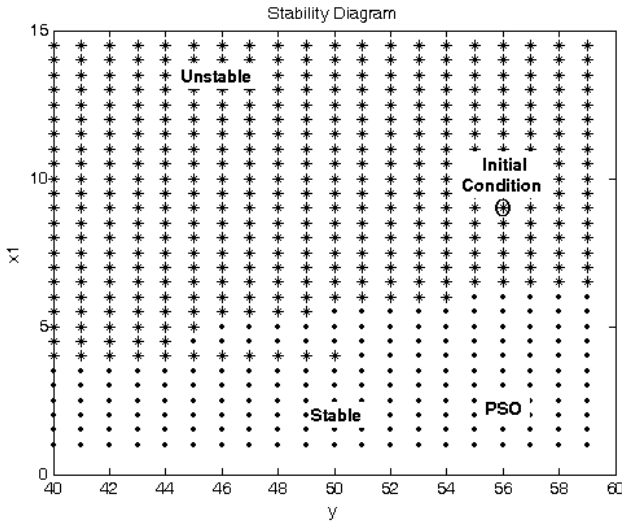


Fig. 4. Stability Diagram for x_j with the region's control applied

3 Swarm Control Design

The particle swarm optimization (PSO), this technique is a population based stochastic optimization technique developed by Kennedy and Eberhart in 1995 [10]. A particle swarm optimization algorithm consists of a number of individuals refining their knowledge of the given search space. In each iteration, the particle swarm optimization algorithm refines its search by attracting the particles to positions with good solutions, considering the best solution until the moment and the best solution of the iteration. The particle swarm optimization technique has ever since turned out to be a competitor in the field of numerical optimization. The PSO approach to nonlinear and control has been observed and discussed in the literature [11-13].

Here, we propose a method for control of unstable systems using the Particle Swarm Optimization with optimization techniques. The method, is used for control the unstable movement of MEMS Gyroscope to stabilize the system to period orbit. The proposed method formulates the nonlinear system identification as an optimization problem in parameter space and then particle swarm optimization are used in the optimization process to find the estimation values of the parameters.

3.1 Particle Swam Optimization

The Particle Swarm Optimization (PSO) algorithm, introduced by Kennedy and Eberhart [10], is a computational simulation of social and biological inspired algorithm.

PSO consists of a algorithm with low computational cost and information sharing innate to the social behavior of the composing individuals. These individuals, also called particles, flow through the multidimensional search space looking for feasible

solutions of the problem. The position of each particle in this search space represents a possible solution whose feasibility is evaluated using an objective function.

The PSO algorithms refines in each iteration, its search by attracting the particles to positions with good solutions, using the best solution (\vec{p}_i) found by the particle in last iteration and the best solution found so far considering all the particles (\vec{p}_g).

In each iteration, a particle i having position \vec{x}_i have its velocity \vec{v}_i updated in the following way:

$$\vec{v}_i = X(w\vec{v}_i + \vec{\varphi}_{1i}(\vec{p}_i - \vec{x}_i) + \vec{\varphi}_{2i}(\vec{p}_g - \vec{x}_i)) \tag{5}$$

where X is know as the constriction coefficient described in [14], w is the inertia weight, \vec{p}_i is best solution found by the particle in last iteration and the \vec{p}_g best solution found so far considering all the particles, and $\vec{\varphi}_1$ and $\vec{\varphi}_2$ are random values different for each particle and for each dimension. The position of each particle is updated during the execution of iteration. This is done by adding the velocity vector to the 1 position vector, i.e.,

$$\vec{x}_i = \vec{x}_i + \vec{v}_i \tag{6}$$

Setting for the velocity parameters determine the performance of the particle swarm optimization to a large extent. This process is repeated until the desired result is obtained or a certain number of iterations is reached or even if the solution possibility is discarded.

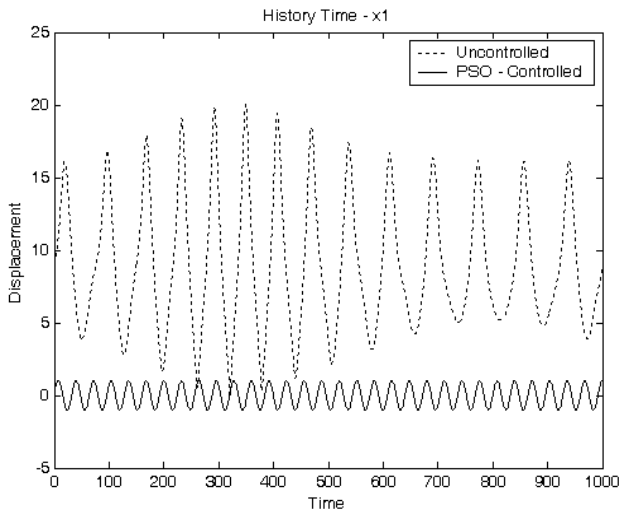


Fig. 5. PSO-Controlled and non-controlled time history x_1

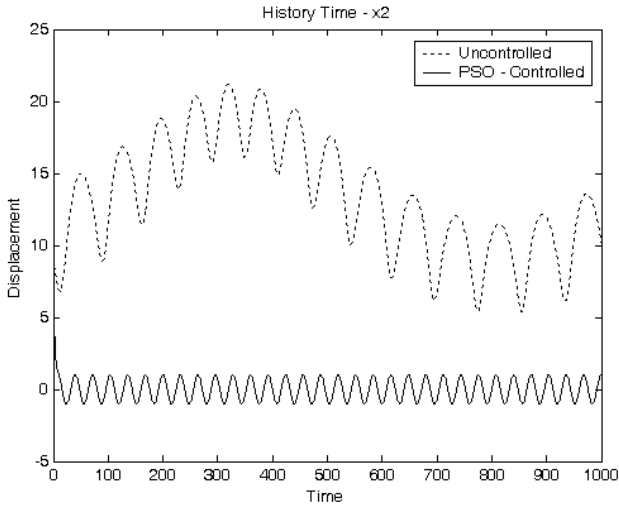


Fig. 6. PSO - Controlled and non-controlled time history x_2

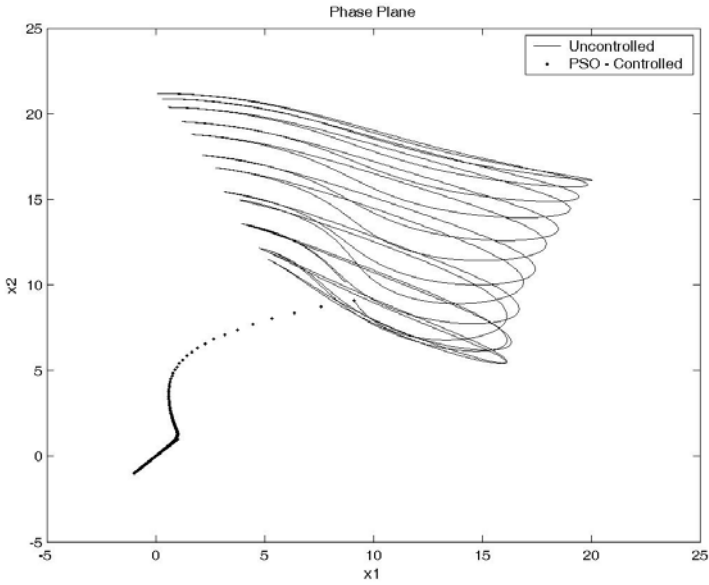


Fig. 7. Phase portrait: PSO controlled and non-controlled

4 Control Swarm Approach

The proposed algorithm, showed above, formulates the nonlinear system identification as an optimization problem in parameter space, and then adaptive particle swarm optimizations are used in the optimization process to find the estimation values of the parameters.

The algorithm is used for control the behavior unstable of the nonlinear dynamics model (4). The goal of this control synthesis is find the estimation values of the parameters, to drive the orbit of the system to a periodic orbit.

We apply the Particle Swarm Optimization algorithm, presented in earlier section for the MEMS Gyroscope (4), to reduce the unstable behavior of this nonlinear system to a period orbit. The Fig. 5, 6 and 7 showed the behavior controlled and uncontrolled of the system (4). In comparing, non-controlled system (see Fig. 2 and 3) with of numerical results of PSO (Fig. 5-7) we can verify that control orbit generated by PSO approach has small diameter.

Algorithm of the Particle Swarm Optimization

Create and initialize an n_x -dimensional swarm, S , through the system of equations (4) and shown in the projection of the phase space (figure 3).

$S.x_i$ i is used to denote the position of particle i in swarm S .

$S.y_i$ i is used to denote the best position of particle i in swarm S .

$S.\hat{y}$, is used to denote the global best position of particle i in swarm S .

```

repeat
  for each particle  $i = 1, \dots, S.ns$  do
    // set the personal best position
    if  $f(S.x_i) < f(S.y_i)$  then
       $S.y_i = S.x_i$ ;
    end
    // set the global best position
    if  $f(S.y_i) < f(S.\hat{y})$  then
       $S.\hat{y} = S.y_i$ ;
    end
  end
  for each particle  $i=1, \dots, S.ns$  do
    update the velocity using equation (5);
    update the position using equation (6);
  end
until stopping condition is true;

```

5 Conclusion

In this work, a dynamics of the MEMS gyroscope, proposed by [5, 6] it is investigated.

We applied the particle swarm optimization technique applied to control MEMS Gyroscope. This control allows reduction of the oscillatory movement of the system to a desired period orbit.

In comparing of numerical results of PSO with the non controlled system (Fig. 2 and 3) we can verify that control orbit generated by PSO approach (Fig. 5-7) has small diameter.

The particle swarm optimization technique presents a computational algorithm motivated by a social analogy. The algorithm control allowed reducing the oscillatory movement of the nonlinear systems to a period orbit. The Fig 5-7, illustrate the effectiveness of the control algorithm to these problem.

Acknowledgments

The first author thanks all the support of the Fundação Hermino Ometto and program of postdoctoral from the State University of São Paulo at Rio Claro.

The second author thanks Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) and Conselho Nacional de Pesquisas (CNPq) for a financial supports.

References

1. Illic, B., Czaplewki, D., Craighead, H.G., Neuzal, P., Campagnolo, C., Batt, C.: Mechanical resonant immunospecific biological detector. *Applied Physics Letters* 77, 450–452 (2000)
2. Stowe, T.D., Yasumura, K., Kenny, T.W., Botkin, D., Wago, K., Rugar, D.: Attonewton force detection using silicon cantilevers. *Applied Physics Letters* 71, 288–290 (1997)
3. Kenny, T.: Nanometer-scale force sensing with MEMS devices. *IEEE Sensors Journal* 1, 148–157 (2001)
4. Rugar, D., Yannoni, C.S., Sidles, J.A.: Mechanical detection of magnetic resonance. *Nature* 360, 563–566 (1992)
5. Miller, N.J., Shaw, S.W., Oropeza-Ramos, L.A., Turner, K.L.: Analysis of a Novel MEMS Gyroscope Actuated By Parametric Resonance. In: *Proceedings of the Sixth EUROMECH Nonlinear Dynamics Conference (ENOC 2008)*, Saint Petersburg, Russia, June 30 – July 4 (2008)
6. Oropeza-Ramos, L., Burgner, C.B., Olroyd, C., Turner, K.: Inherently robust micro gyroscope actuated by parametric resonance. In: *IEEE International Conference on Micro Electro Mechanical Systems*, Tucson, AZ, pp. 872–875 (2008)
7. Ott, B., Grebogi, C., Yorke, J.A.: Controlling Chaos. *Phys. Rev. Lett.* 66, 1196 (1990)
8. Sinhá, S.C., Henrichs, J.T., Ravindra, B.A.: A General Approach in the Design of active Controllers for Nonlinear Systems Exhibiting Chaos. *Int. J. Bifur. Chaos* 165, 10–11 (2000)
9. Rafikov, M., Balthazar, J.M.: On control and synchronization in chaotic and hyperchaotic systems via linear feedback control. *Communications on Nonlinear Science and numerical Simulations* (2007) (in press), doi:10.1016/j.cnsns.2006.12.011

10. Kennedy, J., Eberhardt, R.: Swarm Intelligence. In: Proceeding of IEEE International Conference on Neural Network, pp. 1942–1948 (1995)
11. Hou, Z.: Hammerstein Model Identification Based on Adaptive Particle Swarm Optimization, iita. In: Workshop on Intelligent Information Technology Application IITA 2007, pp. 137–140 (2007)
12. Clerc, M., Kennedy, J.: The particle swarm: explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* 6(1), 68–73 (2002)
13. Chavarette, F.R., Guilherme, I.R.: On Particle Swam Optimization (PSO) Applied to a Micro-Mechanical Oscillator Model. In: Proceeding International Conference on Computational Intelligence for Modelling, Control and Automation (2008), doi:10.1109/CIMCA.2008.31
14. Zahng, W., Baskaran, R., Turner, K.: Nonlinear Behavior of a Parametric Resonance-Based Mass Sensor. In: Proceedings of ASME International Mechanical Engineering Congress & Exposition, pp. 1–5 (2002)

A Representation to Apply Usual Data Mining Techniques to Chemical Reactions

Frank Hoonakker^{1,3}, Nicolas Lachiche², Alexandre Varnek³,
and Alain Wagner^{3,4}

¹ Chemoinformatics laboratory, University of Strasbourg, France

² LSIIT, University of Strasbourg, France

³ eNovalys, Illkirch, France

⁴ Functional ChemoSystems, University of Strasbourg, France

Abstract. Chemical reactions always involve several molecules of two types, reactants and products. Existing data mining techniques, eg. Quantitative Structure Activity Relationship (QSAR) methods, deal with individual molecules only. In this article, we propose to use Condensed Graph of Reaction (CGR) approach merging all molecules involved in a reaction into one molecular graph. This allows one to consider reactions as pseudo-molecules and to develop QSAR models based on fragment descriptors. Here ISIDA fragment descriptors calculated from CGRs have been used to build quantitative models for the rate constant of S_N2 reactions in water. Three common attribute-value regression algorithms (linear regression, support vector machine, and regression trees) have been evaluated.

1 Introduction

Quantitative Structure Activity Relationship (QSAR) consists in predicting some chemical property given the structure of the molecule. It is an important research area in chemistry, and a very challenging application domain for data mining. QSAR typically deals with a single molecule. Chemical reactions usually involve several molecules. As it is possible to predict properties of molecules, the same should be possible with reactions. The problem is to plug several molecules, reactants and products, in a data mining algorithm.

This article points out the use of a Condensed Graph of Reaction (CGR) to represent a reaction involving several molecules as if it was a single molecule, therefore allowing the use of existing techniques dealing with a single molecule. This is illustrated on a real chemical problem.

Chemistry, in particular QSAR, is a main application domain of machine learning and data mining. Inductive Logic Programming and Relational Data Mining can represent and learn from complex structures such as molecules. Moreover they can use background knowledge such as rings, generic atoms [\[1,2,3,4\]](#). However to the best of our knowledge they have not been applied to chemical reactions.

Some papers related to data mining methods predicting properties of reactions have been published, but they do not really model the reaction. For instance

Brauer [5] and Katriski [6] have published papers dealing with Quantitative Structure Reactivity Relationship concerning only one reaction making some parameter (such as solvent) vary. Another attempt has been proposed by Halberstam [7] to model the rate constant of reaction involving two reactants and one product ($A + B \rightarrow C$) where the second reactant (B) is always the same. The study was then reduced to a classical QSAR on one compound.

This paper is organised as follows. The condensed graphs of reactions are defined in section 2. ISIDA fragment descriptors are presented in section 3. The prediction of the rate constant of reaction is described in section 4. Section 5 concludes.

2 Condensed Graphs of Reactions

A Condensed Graph of Reaction [8] represents a superposition of reactants and products graphs. A CGR is a complete connected and non oriented graph in which each node represents an atom and each edge a bond. CGR uses both conventional bonds (single, double, aromatic, etc.) which are not transformed in the course of reaction, and dynamical bonds corresponding to those created, broken or modified during the reaction (cf. Figure 1).

Actually a CGR is a pseudo molecule in which some new bond types have been added. An editor of CGR has been added to our software environment specialised in chemical data mining: ISIDA (In Silico Design and Analysis) [9]. Any new type of dynamical bond could be easily added in the list of bond types.

Moreover we developed an algorithm that generates a CGR from the file formats (RXN and RD) usual in chemoinformatics to model reactions. This programs requires the information about the atom mapping in reactants and products. This information is often available from existing software to edit and manage chemical data, otherwise it can be added thanks to our editor. Indeed

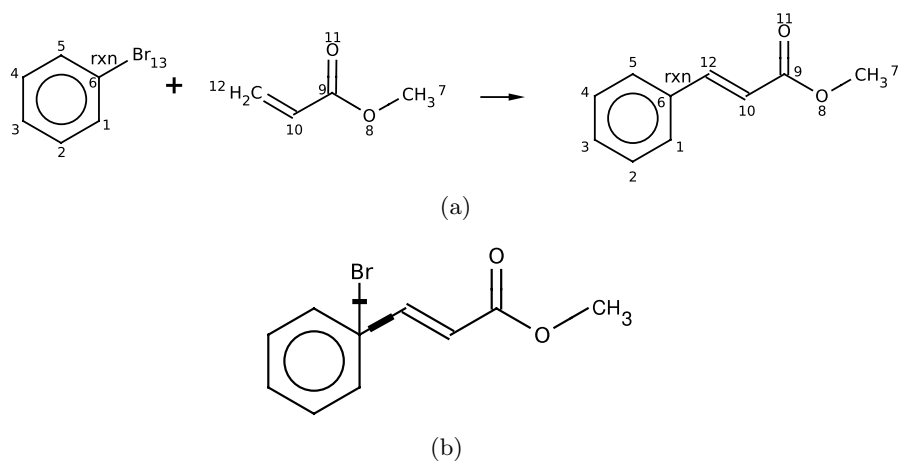


Fig. 1. A reaction (a) and the corresponding Condensed Graph of Reaction (b)

the key point to produce a CGR is to map the reaction, that means that each atom on the left of the arrow corresponds to an atom on the right of the arrow. On Figure 1(a) each atom is uniquely numbered in order to assign the same number to the same atom on both sides of the arrow, for instance the atom numbered 12 on Figure 1(a). Moreover, some flags are added to describe the bonds that change, as described in the "CTFile format" document [10] from Elsevier MDL©. In the case of our example reaction a "rxn" flag is drawn beside the created bond between the atoms mapped 6 and 12 and for the broken bond between atoms 6 and 13.

In most of the database, the mapping is automatically done, with some errors due to mismatching of the atoms on each side of the arrow. For our dataset, the mapping was manually done and verified by a chemist to guarantee avoiding mismatch. Once the reactions are correctly mapped, the CGR are created. The algorithm consists in gathering the atoms of all the compounds of the reaction without duplication of the mapped atom. Then the connection table of reactants and products are examined to find the reactivity flag and write the dynamical bond in the CGR. Figure 1(b) shows the CGR corresponding to the reaction above. Let us emphasize that the bond types assigned between the carbon 6 and the bromine 13 denotes a broken single bond and the bond type between carbons 6 and 12 denotes the creation of a single bond.

This change of representation allows one to store the reaction database in the format (SD) usual in chemoinformatics to represent individual molecules.

3 ISIDA Fragment Descriptors

For each compound, the ISIDA fragment descriptors [11][12][13] produce a vector of integers counting the occurrences of molecular fragments. The nature of each descriptor is a molecular fragment, as detailed below, and its value is the count of this fragment in a molecule.

Fragments are built by computing the shortest paths in the molecular graph between two atoms, in terms of the number of nodes passed through. The fragment is a representation of the Atoms and Bonds (AB) traversed by this path.

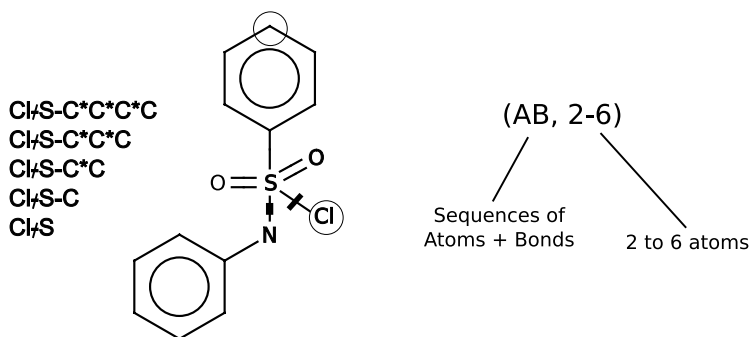


Fig. 2. Example of ISIDA fragment descriptors applied to a Condensed Reaction Graph

The ISIDA fragment descriptors apply to the CGR. For the reactions only fragments containing at least one dynamical bond are selected. Some example of fragments in their linear notation are shown in Figure 2. The first example (Cl/S-C*C*C*C) represents the shortest path between the two atoms circled on the molecular graph (length = 6). If several shortest paths could be found, all of them are taken into account. Symmetric fragments, for example the C-C-N and N-C-C, are considered as a single descriptor. The fragmentation takes a minimum and a maximum length as parameters. In this paper, the fragments having from 2 to 6 atoms were considered I(AB,2-6).

4 Prediction of the Rate Constant of Reaction

This section illustrates the use of the CGR and ISIDA fragment descriptors to predict the rate constant of reaction. First the data are described, then the model.

4.1 Data

The data used for the computation of the rate constant comes from a compilation [14] of the rate and equilibrium constants of heterolytic organic reactions. The selected reactions concern Nucleophile Substitution 2 (SN₂) in water. The database¹ was manually built and contains 249 reactions at 25 Celsius degrees with their $\log(k)$ where k is the rate constant. The $\log(k)$ fluctuates from -6.38 to 4.29, its mean is -1.55 and its standard deviation is 1.84 (cf. Figure 3).

Data in this compilation were extracted from publications implementing various methods and experimental protocols to measure the rate constant. This is the source of some variability in the data set. For instance, different values are reported for the same reaction just by changing the reactant concentration. For instance, for the same reaction, the rate constant can vary in a range of 1 to 1.5

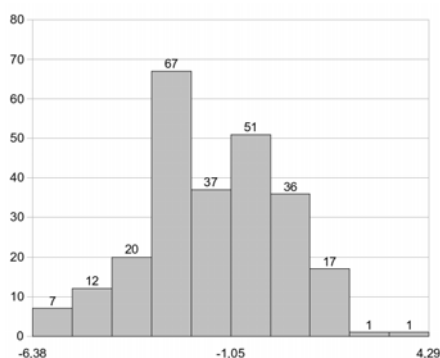


Fig. 3. The repartition of $\log(k)$ for the 249 reactions of the database

¹ Available on demand at Laboratoire d'Infochimie, 4 rue Blaise Pascal 67000 Strasbourg France. varnek-at-infochim.u-strasbg.fr

log units. This might be due to experimental errors. In such situations the mean value has been used as the experimental rate constant. It's not really possible to estimate the experimental error of the data because too many sources are involved. But we estimate our data error around 1 log unit.

4.2 Settings

Three methods were used to model our data : (i) M5P (model tree), (ii) SVM-reg (an SVM method for regression problems) and (iii) linear regression (LR), from *WEKA* [15]. Usually SVM gives more accurate results, but M5P produces models easier to interpret. M5P and the linear regression were used with their default parameters. The SVMreg used a RBF kernel and the default values for its parameters ($c = 1$, $\gamma = 0.01$, $\varepsilon = 0.001$).

The RMSE (Root Mean Squared Error) and the correlation coefficient (R^2) were computed to estimate the error of the procedure. There is sometimes an ambiguity between the correlation coefficient and the determination coefficient. We used the correlation coefficient as defined by the equation [1].

$$R^2 = 1 - \frac{\sum_{i=1}^n (a_i - p_i)^2}{\sum_{i=1}^n (a_i - \bar{a})^2} \quad (1)$$

where p is the predicted value, a the actual value and n the total number of example in the test set. A ten fold cross validation was used for these computations to validate and compare our models. Actually the cross validation procedure was repeated ten times and the result of the calculations for each method are the average of the RMSE and of the correlation coefficients of all the runs. The corresponding standard deviation was evaluated in order to check over that the different splittings do not produce too large variations.

4.3 Results

The performances of the three methods on the I(AB,2-6) fragmentation are reported below:

| | M5P | SVMreg | RR |
|-------|------|--------|------|
| R^2 | 0.64 | 0.73 | 0.59 |
| RMSE | 1.08 | 0.95 | 1.16 |

The correlation coefficient is greater than 0.6. The methods produced models that correctly approximate the constant rate of reaction. Non-linear models (M5P and SVMreg) are better than linear models (RR), and more complex models (SVMreg) are better than the more understandable models produced by M5P. This can be checked on the REC curve, cf. Figure 4. A REC (Regression Error Characteristic) curve [16] plots the accuracy with respect to the error threshold. The accuracy, in the regression context, is defined as the number of instances such that the absolute error between their actual and predicted target values is below the threshold.

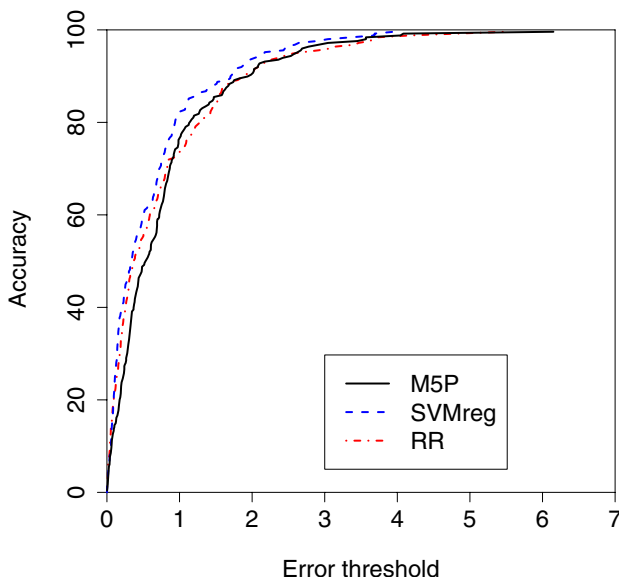


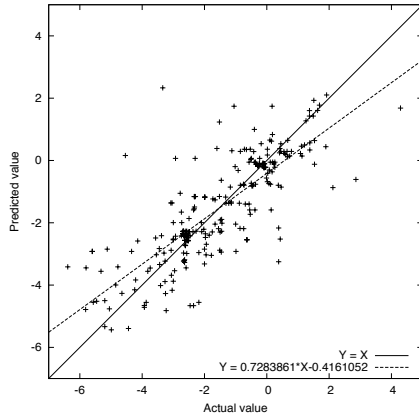
Fig. 4. REC curve (Regression Error Curve) for the three models

Moreover the Root Mean Squared Errors are around 1 log. Let us remind it is the error in the collection of data. It seems difficult to improve the models without improving the data collection.

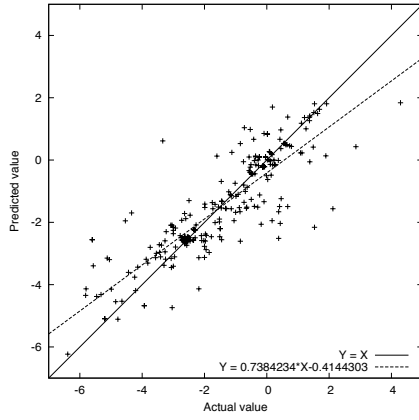
Figure 5 plots the actual and predicted values for each of the 249 reactions. For each model, two lines are added: the identity $Y=X$ in order to illustrate the determination coefficient, and the best linear model fitting the points in order to illustrate the correlation coefficient. RMSE and R^2 measures how much the points are spread around the identity. Points are less spread away with SVMreg than with M5P and RR. Such a plot also allows to identify outliers, points that are further away from the diagonal. Actually most of those points correspond to reactions involving unfrequent fragments, therefore those reactions are more difficult to learn from such a small dataset.

Figure 6 contains the model built by M5P on the whole dataset. For the dynamical bonds, we defined new bond types in MOL files: "5" for the breaking of a single bond, and "1" for the formation of a single bond. The first condition "C-P5F ≤ 0.5 " means that the fragment "a carbon atom with a single bond to a phosphorus with a broken single bond to a fluorine atom" has less than 0.5 occurrences in the CGR. The analyse of this model is focused on the structure of the tree, not on the linear models contained in each leaf, in order to identify whether some leaves correspond to meaningful sets of S_N2 reactions, and at least to check whether the conditions in the nodes make sense.

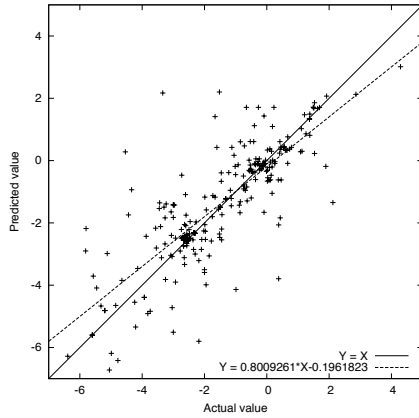
First, the structure of a decision tree is not meaningful. For instance the first leaf (LM1) is defined by the absence of fragments C-P5F, P1O, S-S1C, and Cl5C. Obviously a reaction is defined by the fragment it contains rather



(a) M5P



(b) SVMreg



(c) RR

Fig. 5. Correlation between the actual and predicted values for each model

```

C-P5F <= 0.5 :
|  P10 <= 0.5 :
|  |  S-S1C <= 0.5 :
|  |  |  C15C <= 0.5 : LM1 (35/52.959%)
|  |  |  C15C > 0.5 : LM2 (13/46.886%)
|  |  |  S-S1C > 0.5 : LM3 (24/35.668%)
|  |  P10 > 0.5 :
|  |  |  C-P5S-C-C-C <= 0.5 :
|  |  |  |  P5S-C <= 0.5 :
|  |  |  |  |  C-C-N-P10 <= 1.5 : LM4 (24/44.991%)
|  |  |  |  |  C-C-N-P10 > 1.5 : LM5 (7/50.506%)
|  |  |  |  |  P5S-C > 0.5 :
|  |  |  |  |  |  C-C-O-P10 <= 0.5 : LM6 (8/31.287%)
|  |  |  |  |  |  C-C-O-P10 > 0.5 :
|  |  |  |  |  |  |  P5S-C-C-S-C <= 1.5 :
|  |  |  |  |  |  |  |  S-C-C-S5P-C <= 0.5 :
|  |  |  |  |  |  |  |  |  P5S-C-C-N-C <= 3.5 : LM7 (12/19.83%)
|  |  |  |  |  |  |  |  |  P5S-C-C-N-C > 3.5 : LM8 (4/1.299%)
|  |  |  |  |  |  |  |  |  S-C-C-S5P-C > 0.5 : LM9 (6/0.977%)
|  |  |  |  |  |  |  |  |  P5S-C-C-S-C > 1.5 : LM10 (6/12.731%)
|  |  |  |  |  |  |  |  |  C-P5S-C-C-C > 0.5 : LM11 (34/13.721%)
C-P5F > 0.5 : LM12 (76/44.295%)

```

Fig. 6. A model tree produced by M5P

than by the fragments it does not contain. So we had to find the corresponding reactions in the dataset, and observed that those reactions mainly involve iodine. Reactions involving different halogens have different kinetics. So it is meaningful to distinguish iodine (LM1) from chlorine (LM2). The third leaf (LM3) involves a fragment (S-S1C) that is exceptional, but therefore it is meaningful to isolate the corresponding reactions in a leaf.

We notice that all other branches of the tree, actually 2/3 of the reactions, involve phosphorus. So there is a bias in the dataset. Some branches distinguish whether a bond to fluorine (C-P5F, LM12) or to a sulfur (P5S-C) is broken. Some conditions are refined, for instance P10 is refined into C-C-N-P10 or C-C-O-P10. But some refinements such as P5S-C-C-N-C and P5S-C-C-S-C do not seem meaningful. However they occur at depths 6 and 8 respectively and cover few reactions, hence they might come from overfitting, and they could be avoided by a stronger pruning.

Finally, despite the dataset being too small to finely model all S_N2 reactions, the fragments built from CGR and selected in the model tree make sense.

5 Conclusion

Condensed Graphs of Reactions enable any existing QSAR technique to be applied to chemical reactions. This approach has been successfully experimented on a real chemical problem, using ISIDA fragment descriptors to generate an attribute-value representation of the chemical reactions, and out-of-the-box regression techniques from Weka.

References

1. Dzeroski, S.: Relational Data Mining Applications: An Overview. In: Relational Data Mining. Springer, Heidelberg (2001)
2. Kramer, S., Frank, E., Helma, C.: Fragment generation and support vector machines for inducing sars. SAR and QSAR in Environmental Research 13(5), 509–523 (2002)
3. Helma, C., Cramer, T., Kramer, S., Raedt, L.D.: Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationship of noncongeneric compounds. J. Chem. Inf. Comput. Sci. 44, 1402–1411 (2004)
4. Cannon, E.O., Amini, A., Bender, A., Sternberg, M.J.E., Muggleton, S.H., Glen, R.C., Mitchell, J.B.O.: Support vector inductive logic programming outperforms the naive bayes classifier and inductive logic programming for the classification of bioactive compounds. J. Comput. Aided Mol. Des. 21, 269–280 (2007)
5. Brauer, M., Péres-Lustres, J.L., Weston, J., Anders, E.: Quantitative Reactivity model for the hydration of carbon dioxide by Biometric Zinc Complexes. Inorg. Chem. 41, 1454–1463 (2002)
6. Katritzky, A.R., Perumal, S., Petrukhin, R.: A QSRR Treatment of Solvent Effects on the Decarboxylation of 6-Nitrobenzisoxazole-3-carboxylates Employing Molecular Descriptors. J. Org. Chem. 66(11), 4036–4040 (2001)
7. Halberstam, N.M., Baskin, I.I., Palyulin, V.A., Zefirov, N.S.: Neural networks as a method for elucidating structure-property relationships for organic compounds. Russ. Chem. Rev. 72(7), 629–649 (2003)
8. Fujita, S.: Description of organic reactions based on imaginary transition structures. 1. Introduction of new concepts. J. Chem. Inf. Comput. Sci. 26(4), 205 (1986)
9. Varnek, A.: ISIDA software, <http://infochim.u-strasbg.fr/recherche/isida/index.php>
10. Elsevier MDL: CTfile Format (2007), <http://www.mdli.com/downloads/public/ctfile/ctfile.jsp>
11. Varnek, A., Fourches, D., Hoonakker, F., Solov'ev, V.P.: Substructural fragments: an universal language to encode reactions, molecular and supramolecular structures. J. Comput. Aided. Mol. Des. 19(9-10), 693–703 (2005)
12. Solov'ev, V.P., Varnek, A., Wipff, G.: Modeling of ion complexation and extraction using substructural molecular fragments. J. Chem. Inf. Comput. Sci. 40(3), 847–858 (2000)
13. Todeschini, R., Consonni, V.: Molecular Descriptors for Chemoinformatics. Wiley-VCH (2009)
14. Laboratory of chemical kinetics and catalysis. Tartu State University: Table of rate and equilibrium constants of heterolytic organic reactions (1977)
15. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
16. Bi, J., Bennett, K.P.: Regression Error Characteristic Curves. In: 20th International Conference on Machine Learning (2003)

Incident Mining Using Structural Prototypes

Ute Schmid¹, Martin Hofmann¹, Florian Bader¹, Tilmann Häberle²,
and Thomas Schneider²

¹ Faculty Information Systems and Applied Computer Science,
University of Bamberg, Germany

`firstname.lastname@uni-bamberg.de`

² SAP AG, St. Leon-Rot, Germany

`firstname.lastname@sap.com`

Abstract. Software and other technical products offered to a mass market have a high demand on support and help desks. A tool for automated classification of incident reports, errors and other customer requests which offers previous (successful) hints or solution procedures could efficiently decrease support costs. We propose an approach to mining incidents and other customer requests for support based on generalising structural prototypes from structured data. Retrieval can then be efficiently realised by matching incoming requests against prototypes. We present an application to incident reports in an SAP business information system. Several variants of structure generalisation algorithms were realised and performance for an example test base was evaluated with promising results.

1 Introduction

In many business domains, especially for software companies, dealing with user requests for support and service has a growing demand on time and costs. Such user requests might be concerned with lack of information or understanding of installing or using the software, with the need specialised routines for non-standard problems, or with the report of errors and the need of trouble-shooting strategies. While general recommendations for very frequent requests can be collected on a FAQ site, typically many requests have to be dealt with on an individual basis. If support is distributed between many employees, possibly also distributed over different locations, it can be often the case that one support engineer has to deal with a problem which another support engineer has already solved on a previous occasion. A common data base of user requests and how they were handled (successfully) could reduce time and effort for service and support dramatically. Such a data base could even be the backbone for automated support answers for simple standard requests.

Given a data base with support requests and solution routines, the main problem is to provide a suitable similarity measure for retrieval of a suitable solution routine for a new request. In the context of a case-based reasoning approach (Aamodt & Plaza, 1994), similarity is determined between a new and an already

known case. Alternatively, cases can be generalised into prototypes (Rosch, 1983; Zadeh, 1982; Wilson & Martinez, 1993). In this case, the most similar prototype is retrieved and – depending on the application domain – either the associated standard solution can be applied to the new case or a parametrised solution routine can be instantiated in accordance to the new case. Using prototypes can have an advantage over cases for large data bases because retrieval time can be reduced when new cases have only to be matched against the prototypes and not against all cases. Furthermore, prototype theory (Rosch, 1983) mimics a successful human cognitive strategy (Wiese, Konderding, & Schmid, 2008). A prototype represents the relevant aspects of a set of similar objects or situations while irrelevant details are ignored.

The most prominent approach to similarity in data mining, case-based reasoning and classifier learning is to use feature based measures such as Euclidean distances or other Minkowski metrics (Everitt, Landau, & Leese, 2001). If data are not given as sets of features but in form of a structured representation – e.g., as records or as terms – there are two strategies to obtain a similarity rating available: An obvious approach is to transform the structures into feature sets (Yan, Zhu, Yu, & Han, 2006; Geibel, Schädler, & Wysotzki, 2003). This has the advantage that many standard approaches to data-mining and classification can be used. Another possibility is to use specialised approaches to structural similarity such as edit distance (Bunke & Messmer, 1994) or determining greatest common structures (Messmer & Bunke, 2000; Plaza, 1995; Estruch, Ferri, Hernández-Orallo, & Ramírez-Quintana, 2009). Structure-based approaches have the advantage that information contained in the relations between objects is not lost by transformation into features.

Our area of application is concerned with incident reports for the business information software SAP Business ByDesign. In this software, creation and visualisation of incident reports are based on an incident model. Incident reports are created by an incident wizard. The current system state together with all context data (current workspace, current object, UI) and a filled in report mask are saved into an XML document and sent first to a key user and – if he or she cannot solve the problem – sent further to a support engineer at SAP. The reports can be viewed by the support engineer in the SAP support studio software where he or she has different possibilities to analyse the report using a graphical presentation of context data.

The work presented here, is a first exploration of the utility of structure generalisation in this domain. Currently, we work on manually created incident clusters and focus on generalisation and retrieval. In the following, we first present how incidents are represented in form of trees. Afterwards, we introduce our approach to tree generalisation and retrieval. Within the general framework of structure matching and learning we propose different algorithmic realisations. An evaluation of these realisations and a comparison with the inductive logic programming algorithm FOIL for a set of sample incidents is presented. We conclude with possible improvements, extensions, and suggestions for application.

2 Incident Trees

Incidents occurring in the context of the SAP Business ByDesign System are represented in a unique form, given as an incident model which is specified as an XML-tree. It contains, for example, information about the software version, the workcenter (the role of the user) and the business object for which the incident occurred. The model is an abstraction of the system’s class hierarchy. Incident objects are referred to by a name, their corresponding class is given as a type. The details of the incident model are reported in Bader (2009). The general structure of an incident follows the following form: Each element has a prescribed type which characterises the context information and the content of an incident. For a given incident, an element is instantiated with a name-string. Depending on its type, an element can have zero to a typically small number n of children. An extract of the incident model is given in Figure 1.

An example use case is, that an employee wants to order some office equipment. He or she works from the “home” workcenter ($WC(home)$) and entered the shopping basket user interface ($UI(ShopBaskQAF)$). When an incident is reported in this situation, the business object “purchase request” ($BO(PurReq)$) becomes part of the incident context.

The incident model – also called model tree – represents the general structure which is underlying each possible incident report and thereby restricts the form of incident trees.

Definition 1. A **model tree** M is a tree of fixed size with typed elements $e : \tau$. The type of the element determines the number and types of its child nodes.

Note, that in Figure 1 we write an element e as $\tau(name)$ where $name$ is a variable which can be instantiated by a constant name-string for a given incident.

Definition 2. An **incident tree** I is an instantiation of the model tree M : Each element $e \in M$ is either mapped to ϵ (empty element) or a constant name string of type τ . At least one element in I must be unequal ϵ .

Definition 3. To refer to an **element** e in a tree T , we write e if addressing the element only and we write $e(T_1, \dots, T_n)$ if we address the element and its children. A **position** in a tree T is defined as

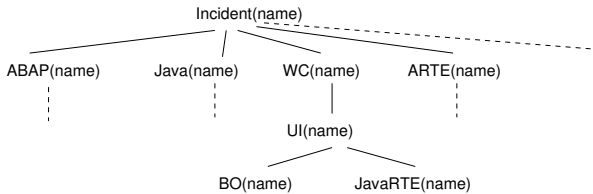


Fig. 1. Extract of the Incident Model

- λ is the root position of T ,
- if $T = e(T_1, \dots, T_n)$ and u is a position in T_i , then $i.u.$ is a position in T .

$T.p = e$ refers to a specific element in T . $T.p = e(T_1, \dots, T_n)$ refers to a specific element and its children.

For better readability we omit types in the definition. In the algorithms presented in the following section, it is guaranteed that mapped elements are of the same type since mapping is guided by a fixed model tree underlying all instance trees.

3 Tree Generalisation and Retrieval

For learning of incident prototypes, sets of incidents – called cluster – are generalised with respect to their common structure. In consequence, each cluster is represented by a prototype and new incoming incidents are compared with all prototypes to retrieve the most similar one. In the following we propose three approaches to generalisation and retrieval: Anti-unification of trees as base-line approach and two variants for generating structure-dominance trees. Since we are not concerned with arbitrary trees but with trees based on a unique structure given as model tree, there is no need to rely on general approaches to tree matching (Wang, Zhang, Jeong, & Shasha, 1994).

3.1 Anti-unification of Trees

Syntactic first-order anti-unification is an approach to generate least generalisations over terms (Plotkin, 1969; Burghardt & Heinz, 1996). An anti-instance of a set of terms is calculated by traversing them simultaneously and keeping the common structure. If terms start with different symbols, these terms are represented as mappings to variables in the anti-instance. The mappings can be used to recreate the original terms from the anti-instance by transforming them into substitutions.

Since trees and terms are corresponding data structures, this approach can be transferred to incident trees. That is, a cluster prototype is defined as anti-instance of a set of incident trees. Instead of calculating mappings, we introduce an ϵ -element in the anti-instance at the position of mismatched terms.

Algorithm 1. *Let M be a model tree, $I = \{I_1, \dots, I_N\}$ a set of incident trees, and P a prototype tree. **Syntactic anti-unification of sets of incident trees** $\text{au}(p, P, M, I)$ is defined as:*

- Initially the prototype is empty: $P.\lambda = \epsilon$.
- We traverse the model tree top-down, starting with $M.\lambda$.
- For the current position p in model tree M , $M.p = e : \tau$
 - If for all incident trees in I holds $I_1.p = \dots = I_N.p$ then $P.p := e$ and if $M.p = e(T_1, \dots, T_n)$ then for all incidents I_i do $\text{au}(p.i, P, M, \{T_{1i}, \dots, T_{Ni}\})$.
 - Else $P.p := \epsilon : \tau$ (an empty element).

An anti-instance of a set of trees corresponds to the intersection of all trees with respect to a model tree (see Figure 2 for an illustration).

For retrieval, the most similar prototype P_i for an incoming incident I_{new} must be determined. This can be realised by anti-unifying I_{new} with each prototype and partially ordering the anti-instances $A_{P_i, I_{new}}$ with respect to their subsumption relation (Plaza, 1995).

Definition 4. *An incident tree T is said to subsume another incident tree T' , that is, T is a generalisation of T' ($T > T'$) if $\text{sub}(T, T', \lambda) = \text{true}$ with*

- $\text{sub}(\epsilon, T', p) = \text{true}$
- $\text{sub}(T, T', p) = \text{false}$ if $T.p \neq \epsilon$ and $T.p \neq T'.p$
- $\text{sub}(T(t_1, \dots, t_n), T'(t'_1, \dots, t'_m), p) = \text{true}$ if $T.p = T'.p$ and $n = m$ and for all t_i $\text{sub}(t_i, t'_i, p.i)$.

3.2 Structure Dominance Tree Generalisation

Syntactic anti-unification is not robust with respect to noise. Furthermore, using an identity criterium for element matching is very strict. If, for example, $n - 1$ incidents have an identical element at position p and only one incident has a different or empty entry for this element, the prototype at this position is empty. Therefore, we introduce a new approach to prototype learning – structure dominance tree generalisation (SDTG). The basic idea is to collect the number of occurrences of different elements at a position.

Algorithm 2. *Let M be a model tree, $I = \{I_1, \dots, I_N\}$ a set of incident trees, and P a prototype tree. **Structure dominance tree generalisation** $\text{sd}tg(p, P, M, I)$ is defined as:*

- *Initially the prototype is empty: $P.\lambda = \epsilon$.*
- *We traverse the model tree top-down, starting with $M.\lambda$.*
- *Until I is empty, for the current position p in model tree M , $M.p = e : \tau$*
 - *If for all incident tree in I holds $I_1.p = \dots = I_N.p$ then $P.p := e : \tau$ and if $M.p = e(T_1, \dots, T_n)$ then for all incidents I_i do $\text{sd}tg(p.i, P, M, \{T_{1i}, \dots, T_{Ni}\})$.*
 - *Else for each of the $c = |\{I_1.p, \dots, I_N.p\}|$ different elements create a new node $P.p_1, \dots, P.p_c$ with $P.p_i := [e_i / f_i] : \tau$ as element names e_i and their relative frequencies f_i . Proceed for all new elements with $\text{sd}tg(p.i, P, M, \{T_{1i}, \dots, T_{Ni}\})$.*

While syntactic anti-unification returns the intersection of a set of incident trees as prototype, SDTG returns the union of all incident trees (see Figure 2 for an illustration).

A combination of both approaches can be realised as follows:

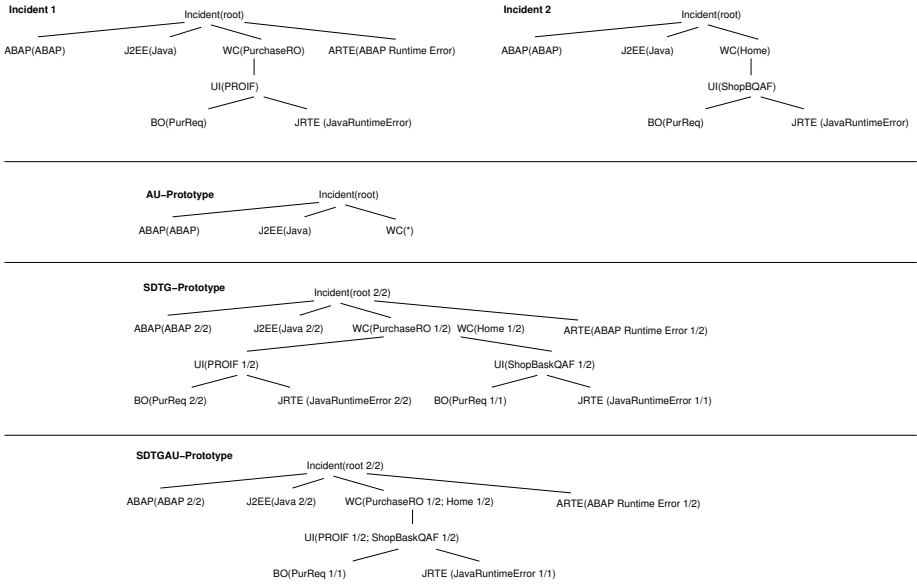


Fig. 2. Illustration of Prototype Generation

Algorithm 3. Let M be a model tree, $I = \{I_1, \dots, I_N\}$ a set of incident trees, and P a prototype tree. **Structure dominance tree generalisation with anti-unification** $\text{sdtgau}(p, P, M, I)$ is defined as:

- Initially the prototype is empty: $P.\lambda = \epsilon$.
- We traverse the model tree top-down, starting with $M.\lambda$.
- Until I is empty, for the current position p in model tree M , $M.p = e : \tau$
 - If for all incident tree in I holds $I_1.p = \dots = I_N.p$ then $P.p := e$ and if $M.p = e(T_1, \dots, T_n)$ then for all incidents I_i do $\text{sdtgau}(p.i, P, M, \{T_{1i}, \dots, T_{Ni}\})$.
 - Else $P.p := [e_i/f_i] : \tau$ becomes a tuple of all occurring elements at position p together with their relative frequencies and we proceed for all children with $\text{sdtgau}(p.i, P, M, \{T_{1i}, \dots, T_{Ni}\})$.

An illustrative example is given in Figure 2.

Retrieval for SDTG and SDTGAU can be realised using some similarity measure over trees. We explored several measures and it showed, that Manhattan distance is the most robust and reliable measure:

$$d(I, P) = \sum_{i=1}^n |f_{Pi} - f_{Ii}| \quad \text{with} \quad f_{Pi} = 1.$$

4 Empirical Results

For empirical evaluation of our structural approaches to incident mining, we obtained 57 real example incidents from SAP support which were manually grouped in 11 clusters based on their root cause analysis. One cluster contained three incidents, four clusters four incidents each, one cluster five incidents, four clusters six incidents each and one cluster 9 incidents. The rather low number of instances per cluster is realistic for this application domain. That is, for this domain only approaches to prototype or classifier learning which produce reliable results for small numbers of cases are applicable. The size of incidents varied between 27 and 2646 nodes with an average size of 812 nodes.

In addition to the approaches described above, we used the inductive logic programming algorithm FOIL (Quinlan & Cameron-Jones, 1995) which is a well established approach to learning from relational data. Since FOIL needs positive and negative examples for its rule induction, for each cluster we used one incident from all other clusters as negative example. Furthermore, for using FOIL the incident model was represented as set of Prolog clauses given as background knowledge.

All evaluations were run on an Intel Core 2 Duo with 2.4 GHz, 2 GB DDR3 working memory, 250 GB hard disk space and operation system Mac OS X 10.5.6. Algorithms were realised with Java Sun JDK 1.5.0_16-b06-284. FOIL in version 6 was compiled with GCC 4.01. Since running times for all trials had a very low standard deviation for prototype generation as well as for retrieval, we only give average run times and omit giving standard deviations.

For a first evaluation, for each of the 11 clusters the prototypes were generated over all incidents (see Table 1). Afterwards, each instance was used in the retrieval phase. Times for generation of prototypes and for retrieval were averaged over all runs. In general, times for all approaches were reasonably fast. As was to be expected, anti-unification returned perfect results with the smallest prototypes (168 nodes in average). FOIL produced a rather large number of erroneous and ambiguous classifications. This result is mostly due to our unsophisticated approach for presentation of negative examples. Since negative examples are used to specialise rules, typically some care in providing suitable negatives is needed. Both SDTG and SDTGAU returned no perfect, but acceptable results.

A more precise evaluation was realised using a leave-one-out approach (see Table 2). Due to the small number of examples, we used three runs for prototype

Table 1. Base Performance: All instances included in prototype generation, all instances used for retrieval

| Method | Hits | Errors | Av. Size | Generation (sec.) | Retrieval (sec.) |
|--------|------|---------------------|----------|-------------------|------------------|
| FOIL | 27 | 18, 12 ¹ | | 0,035 | 0,403 |
| AU | 57 | 0 | 168 | 0,177 | 0,038 |
| SDTG | 53 | 4 | 447 | 0,182 | 0,047 |
| SDTGAU | 55 | 2 | 273 | 0,162 | 0,038 |

¹ first value: classification error, second value: ambiguous result

Table 2. Leave-one-out Performance: Three trials with one instance excluded from prototype generation

| Method | Hits | Errors | Av. Size | Generation (sec.) | Retrieval (sec.) |
|--------|------|--------------------|----------|-------------------|------------------|
| FOIL | 11 | 8, 14 ¹ | | 0,109 | 0,409 |
| AU | 29 | 4 | 173 | 0,485 | 0,044 |
| SDTG | 22 | 11 | 387 | 0,456 | 0,049 |
| SDTGAU | 31 | 2 | 264 | 0,419 | 0,047 |

¹ first value: classification error, second value: ambiguous result

generation disregarding the first, the second and the third incident in each cluster respectively. Again, times are given as averages over all runs. Again, all times for prototype construction and for retrieval are acceptable. Now anti-unification returns the wrong prototype in 4 out of 33 cases which is better than SDTG but slightly worse than SDTGAU.

Finally, we evaluated how our approaches can deal with noisy data. Out of the original 11 clusters, we created 110 new clusters, each containing $n - 1$ incidents of an original cluster and one incident of one of the other clusters (see Table 3). As was to be expected, anti-unification breaks down for noisy data. SDTGAU outperformed all other approaches returning only 1 misclassification.

Table 3. Performance on noisy clusters: 110 clusters each containing one miss-placed instance

| Method | Hits | Errors | Av. Size | Generation (sec.) | Retrieval (sec.) |
|--------|------|--------------------|----------|-------------------|------------------|
| FOIL | 0 | 0, 57 ¹ | | 0,020 | 7,378 |
| AU | 4 | 53 | 66 | 0,202 | 0,051 |
| SDTG | 52 | 5 | 619 | 0,227 | 0,102 |
| SDTGAU | 56 | 1 | 410 | 0,188 | 0,069 |

¹ first value: classification error, second value: ambiguous result

5 Conclusions and Further Work

For the given application domain – mining of incident reports characterised by an incident model – we could show first promising results for a set of simple structure-generalisation algorithms. Of course, the number of clusters and incidents used in the evaluation is rather small and a further evaluation using a larger data base should be realised. Using a small set of incidents had the advantage that we had full control over clustering which was done manually by a domain expert. For a large scale performance evaluation, we need to extend our approach to automated clustering as a first step.

For clustering we again propose to take into account the incident structures. That is, we plan to realise a clustering based on structural similarity between instances (Taskar, Segal, & Koller, 2001).

The proposed algorithms do not take into account the possibility that a tree element might have more than one child of the same type. A planned extension

of the algorithms therefore is, to include tree matching to obtain best matches for arbitrary sets of type-identical nodes.

After extension of our approach to automated clustering, we plan to realise a semi-automated assistance tool for support engineers with SDTGAU as generalisation algorithm: For an incoming incident, a ranked list of retrieved prototypes can be offered. If the engineer accepts one of these prototypes, the new incident is saved in the selected cluster. The support engineer furthermore can edit the prescribed support routines associated with the cluster prototypes. In repeated intervals – after substantial growth of the incident data base – automated clustering and prototype generalisation can be re-done to stratify the data-base. We assume that such a tool might relieve support engineers of repetitive work, considerably heighten efficiency of support and ultimately provide fast and reliable support for the users.

References

- Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications* 7(1), 39–59 (1994)
- Bader, F.: Model-based classification of incident reports in a business information system. Sn approach to prototype-learning by structure generalisation (in German), Unpublished master's thesis, University of Bamberg (2009)
- Bunke, H., Messmer, B.T.: Similarity measures for structured representations. In: Wess, S., Richter, M., Althoff, K.-D. (eds.) *EWCBR 1993. LNCS*, vol. 837, pp. 106–118. Springer, Heidelberg (1994)
- Burghardt, J., Heinz, B.: Implementing anti-unification modulo equational theory (vol. 1006; Tech. Rep.). *Arbeitspapiere der GMD* (1996)
- Estruch, V., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.: Defining inductive operators using distances over lists. In: Schmid, U., Kitzelmann, E., Plasmeijer, R. (eds.) *Proceedings of the 3rd Workshop on Approaches and Applications of Inductive Programming (AAIP 2009)*, pp. 41–64. Edinburgh (2009)
- Everitt, B.S., Landau, S., Leese, M.: *Cluster analysis*, 4th edn. Wiley, Chichester (2001)
- Geibel, P., Schädler, K., Wysotzki, F.: Connectionist construction of prototypes from decision trees for graph classification. *Intelligent Data Analysis* 7(2), 125–140 (2003)
- Messmer, B.T., Bunke, H.: Efficient subgraph isomorphism detection: A decomposition approach. *Efficient Subgraph Isomorphism Detection: A Decomposition Approach* 12(2), 307–323 (2000)
- Plaza, E.: Cases as terms: A feature term approach to the structured representation of cases. In: Aamodt, A., Veloso, M.M. (eds.) *ICCBR 1995. LNCS*, vol. 1010, pp. 265–276. Springer, Heidelberg (1995)
- Plotkin, G.D.: A note on inductive generalization. In: Plotkin, G.D. (ed.) *Machine intelligence*, vol. 5, pp. 153–163. Edinburgh University Press (1969)
- Quinlan, J., Cameron-Jones, R.: Induction of logic programs: FOIL and related systems. *New Generation Computing, Special Issue on Inductive Logic Programming* 13(3-4), 287–312 (1995)
- Rosch, E.: Prototype classification and logical classification: The two systems. In: Scholnick, E. (ed.) *New trends in conceptual representation: Challenges to Piaget's theory?*, pp. 73–86. Lawrence Erlbaum, Hillsdale (1983)

- Taskar, B., Segal, E., Koller, D.: Probabilistic clustering in relational data. In: Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 2001), pp. 870–887 (2001)
- Wang, J.T.L., Zhang, K., Jeong, K., Shasha, D.: A system for approximate tree matching. *IEEE Transactions on Knowledge and Data Engineering* 6(4), 559–571 (1994)
- Wiese, E., Konerding, U., Schmid, U.: Mapping and inference in analogical problem solving – As much as needed or as much as possible? In: Love, B., McRae, K., Sloutsky, V.M. (eds.) *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pp. 927–932. Lawrence Erlbaum, Mahwah (2008)
- Wilson, D.R., Martinez, T.R.: The potential of prototype styles of generalization. In: *Proceedings of the Sixth Australian Joint Conference on Artificial Intelligence (AI 1993)*, pp. 356–361 (1993)
- Yan, X., Zhu, F., Yu, P.S., Han, J.: Feature-based similarity search in graph structures. *ACM Transactions on Database Systems* 31(4), 1418–1453 (2006)
- Zadeh, L.: A note on prototype theory and fuzzy sets. *Cognition* 12, 291–297 (1982)

Viability of an Alarm Predictor for Coffee Rust Disease Using Interval Regression^{*}

Oscar Luaces¹, Luiz Henrique A. Rodrigues², Carlos Alberto Alves Meira³,
José R. Quevedo¹, and Antonio Bahamonde¹

¹ Artificial Intelligence Center. Universidad de Oviedo at Gijón, Asturias, Spain
www.aic.uniovi.es

² Universidade Estadual de Campinas, Faculdade de Engenharia Agrícola,
Cx. P. 6011, CEP 13083-875 Campinas, SP, Brazil

³ Embrapa Informática Agropecuária, Cx. P. 6041,
CEP 13083-970 Campinas, SP, Brazil

Abstract. We present a method to formulate predictions regarding continuous variables using regressors able to predict intervals rather than single points. They can be learned explicitly using the so-called insensitive zone of regression Support Vector Machines (SVM). The motivation for this research is the study of a real case; we discuss the feasibility of an alarm system for coffee rust, the main coffee crop disease in the world. The objective is to predict whether the percentage of infected coffee leaves (the incidence of the disease) will be above a given threshold. The requirements of such a system include avoiding false negatives, seeing as these would lead to not preventing the disease. The aim of reliable predictions, on the other hand, is to use chemical prevention of the disease only when necessary in order to obtain healthier products and reductions in costs and environmental impact. Although the breadth of the predicted intervals improves the reliability of predictions, it also increases the number of uncertain situations, i.e. those whose predictions include incidences both below and above the threshold. These cases would require deeper analysis. Our conclusion is that it is possible to reach a trade-off that makes the implementation of an alarm system for coffee rust disease feasible.

1 Introduction

In this paper we discuss how to learn alarm functions in a real world problem. Starting from a faithful description of present circumstances, these functions must predict future situations of risk so that we may then act to prevent any foreseeable damage. In this context, the costs of prediction errors are not symmetrical: the consequences of false prediction of an alarm (*false positives*) are often not as serious as those of predicting false non-alarms (*false negatives*).

^{*} The research reported here is supported in part under grant TIN2008-06247 from the MICINN (Ministerio de Ciencia e Innovación, of Spain), and grant 2009/07366-5 from the FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo, Brazil).

We deal with continuous target variables whose values above a given threshold should be notified as soon as possible with the highest degree of accuracy. We try, at least, to minimize the percentage of false negatives. The straightforward approach is to learn a regressor: in addition to providing alarm alerts, the regressor produces a numeric assessment of how serious the situation may be.

We present an agriculture case study. The incidence of coffee rust epidemics is caused by a fungus called *Hemileia vastatrix* Berk. & Br., a devastating disease to coffee plantations. This incidence can be measured by the percentage of leaves infected by the fungus. It is well known that the factors that stimulate the growth of the fungi are weather conditions, the type of plantation and the current incidence. Thus, a regression learning task must include these features as predictors and the future incidence as the target value.

We trained a Regression Support Vector Machine (*SVM*) with quite good results. The correlation between predicted and actual incidences is about 0.94 in a cross-validation experiment. However, if we try to devise an alarm system for predicting values above a given threshold, we find that the number of false negatives is too high. In this case the threshold is $\tau = 4.5$. This is not an academic parameter, it is the threshold used in Brazilian plantations; see [6,7,8] for a detailed discussion.

To overcome this weakness of regression, we try to learn models to predict *approximations* to incidence values instead of exact values. To implement this idea, there are a number of possible alternatives. In the approach presented here, we relax the specifications of regression, changing target points for intervals of a fixed width, say 2ϵ . Following [1,5], these predictors may be called *nondeterministic* regressors.

The method employed to learn intervals of fixed width uses regression SVM. These learning algorithms search for predictors that minimize a loss function which ignores errors situated within a certain distance (ϵ) of the true value: ϵ -insensitive loss functions.

To transform interval predictions into alarms, we adopt a cautious policy. Only those intervals completely included below the threshold will be understood as non-alarms. On the other hand, if a predicted interval is above the threshold, that will mean an alarm. However, we have a third possibility situated somewhere in between: predicted intervals that include points above and below the threshold. We label these situations as *warnings*. The usefulness of warnings is that they capture classification errors of pure deterministic regressors. In fact, these errors arise for predictions near the threshold.

In other words, we can convert misclassifications into a type of situation that may require deeper analysis. However, when the alarm system predicts an alarm, and especially a non-alarm, the confidence in these predictions is very high. The radius ϵ of the intervals is proportional to the number of warnings and hence to the prudence of the whole alarm system. In this sense, our approach is closely related to that of classifiers with a reject option [2,4].

Our conclusion is that a trade off between the number of false negatives and warnings would lead to a useful alarm system for coffee rust. The search for an

optimal value for ϵ is beyond the scope of this paper, as we would have to consider the important economic and environmental aspects involved in coffee growing. Nonetheless, considering the results reported at the end of the paper (Section 6), the feasibility of implementing an alarm system is guaranteed. Moreover, the only requirement is a cheap weather station.

In the next section the coffee rust disease and the dataset used in the paper are presented in detail. Sections 3 and 4 are devoted to deterministic and non-deterministic alarms respectively. In Section 5 we discuss the temporal perspective of the alarms.

2 Coffee Rust

The coffee rust caused by fungi *Hemileia vastatrix* Berk. & Br. is the main coffee crop disease in the world. In Brazil, damages lead to yield reduction of up to 35% in regions where climate conditions are propitious to the disease. The impact is thus considerable due to the economic importance of coffee crop.

The traditional way to prevent the disease is to apply agrochemical fungicides on fixed calendar dates. However, the fungicides contaminate the environment and reduce the quality of the coffee. Moreover, as the intensity of the disease between seasons suffers major variations, the use of agrochemicals is not always justified.

The aim of this paper is to discuss the viability of building alarm functions to alert on high incidences of coffee rust. The purpose would be to build economically viable control measures. Our proposal is a predictor, learned using data mining tools, that would allow applying agrochemicals only when necessary, leading to healthier products and reductions in costs and environmental impact.

It is important to emphasize here that fungicides must be applied in advance since they need several days to take effect in coffee plants. Having all this in mind, we used a dataset [6,7,8] whose temporal dimension is very important. The data was obtained on a monthly basis from an experimental farm (Fundação Procafé, Varginha, MG, Brazil), from October 1998 to October 2006, with reports of coffee rust incidences. In September of each year (beginning of agricultural season), eight plots producing coffee were selected, four with thin spacing (approximately 4000 plants/ha) and four with dense spacing (approximately 8000 plants/ha). For each case, two plots were selected with high fruit load (above 1800 kg/ha) and two with low fruit load (below 600 kg/ha). There was no disease control in those plots. Meteorological data was automatically registered every 30 minutes by a weather station close to where the incidence of coffee was being evaluated.

2.1 The Learning Task

From a formal point of view, throughout this paper we deal with the dataset described as follows.

Let \mathcal{X} be a set of descriptions of current situations. Here we wanted to represent, using the data collected, the idea that an alarm system can be used at

any time, not only from the first day of one month to guess the incidence in the first day of the next month. In the coffee rust problem, if we want to predict the incidence of the fungi in a *target* day, we consider predictions made with different days ahead. Thus \mathcal{X} is a set of vectors whose components are:

- Fruit load of the plantation: low (1) or high (2)
- Spacing between plants: dense (1) or thin (2)
- Percentage of leaves infected by fungi in date d_0
- Days from d_0 till now (the day we make the prediction)
- Days from now till the target day: 1 month, 25, 20, 15 and 10 days
- Weather scores in the last 45 days

The weather scores are 13 variables per day, and they include: temperatures, solar radiation, number of hours with sun light, wind speeds, rain, relative humidity, number of hours with relative humidity above 95%, average temperature during these hours, and the same values but during the night. For more details, see [78].

Therefore, the dimension of the vectors of the input space \mathcal{X} is 590. On the other hand, the output space in this case is just the interval of real numbers, $\mathcal{Y} = [0, 100]$, to capture the percentage of coffee leaves infected by the fungi.

3 Regression and Deterministic Alarms

We start presenting the baseline approach obtained from a standard regression tool. From a formal point of view, learning tasks can be presented in the following general framework. Let \mathcal{X} be an input space, and let \mathcal{Y} be an output space. A *learning task* is given by a training set $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ drawn from an unknown distribution $Pr(X, Y)$ from the product $\mathcal{X} \times \mathcal{Y}$. The aim of such a task is to find a hypothesis h (of a space \mathcal{H} of functions from \mathcal{X} to \mathcal{Y}) that optimizes the *expected prediction performance (or risk)* on samples independently and identically distributed (i.i.d.) according to the distribution $Pr(X, Y)$:

$$R^\Delta(h) = \int \Delta(h(\mathbf{x}), y) d(Pr(\mathbf{x}, y)), \quad (1)$$

where $\Delta(h(\mathbf{x}), y)$ is a loss function that measures the penalty due to the prediction $h(\mathbf{x})$ when the true value is y .

If \mathcal{Y} is a metric space (usually the set of real numbers), the learning job is a *regression* task, as in the case of coffee rust. In this case, the aim of learners is to obtain a hypothesis whose predictions are as similar as possible to actual values in the output space. This can be accomplished, for instance, using least squares regression.

On the other hand, the goal of SVM regressors is to minimize the so-called ϵ -*insensitive* loss function. If ϵ is a positive value, this loss does not penalize predictions whose distance to true values is below ϵ ; in symbols,

$$\Delta_\epsilon(h(\mathbf{x}), y) = \max\{0, |h(\mathbf{x}) - y| - \epsilon\}. \quad (2)$$

In any case, once we have learned a regressor h , if τ is a threshold in \mathcal{Y} , we interpret the outputs of h as follows

$$Alarm(h(\mathbf{x})) = \begin{cases} \text{non-alarm} & h(\mathbf{x}) \in (-\infty, \tau] \\ \text{alarm} & h(\mathbf{x}) \in (\tau, +\infty). \end{cases} \tag{3}$$

Notice that the performance of what has been learned can be measured in two different but complementary ways: using the scores of regressors applied to h , and the scores of classifiers applied to $Alarm \circ h$.

4 Regression with Broad Insensitive Zone: Nondeterministic Alarms

Let us assume that we have a regressor whose accuracy to predict a continuous variable is not completely satisfactory. For instance, the performance of a regressor may fail when it is measured in terms of alarm classifications (Eq. 3). This is the case of regressors obtained from the coffee rust learning task. The scores will be discussed later in Section 6. To overcome this problem, as was explained in the introduction, we are going to use regressors allowed to predict intervals rather than single points.

The idea is that the true class of an entry \mathbf{x} may be *somewhere* into the predicted interval for \mathbf{x} . A simple way to implement this idea is to look for regressors that predict intervals of a fixed width, say 2ϵ . Notice that this is exactly the semantics of ϵ -insensitive zone (Eq. 2). For later reference, we recall the formulas of SVM regressors here.

Given a regression learning task S (Section 3) and a *tube* value $\epsilon > 0$, a regression SVM learns a function

$$h_\epsilon(\mathbf{x}) = \sum_{i=1}^n (\alpha_i^- - \alpha_i^+) K(\mathbf{x}_i, \mathbf{x}) + b^*, \tag{4}$$

where K is the *rbf* kernel, $K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$; b^* , and α^+ , α^- are respectively the solution and the Lagrange multipliers of the following convex optimization problem:

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^n (\xi_i^+ + \xi_i^-), \\ \text{s.t.} \quad & (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) - y_i \leq \epsilon + \xi_i^+, \quad y_i - (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) \leq \epsilon + \xi_i^-, \\ & \xi_i^+, \xi_i^- \geq 0, \quad i = 1, \dots, n. \end{aligned} \tag{5}$$

The interval regressor associated to h_ϵ is then defined by

$$h_{ND(\epsilon)}(\mathbf{x}) = [h_\epsilon(\mathbf{x}) - \epsilon, h_\epsilon(\mathbf{x}) + \epsilon]. \tag{6}$$

Notice that we have one optimal interval regressor for each value of the tube, ϵ . The regressor h_ϵ , accordingly to (Eq. 5) is different for each value of ϵ . When

the aim is to learn a deterministic regressor, typically ϵ is a small number; by default, we use $\epsilon = 0.1$. However, for interval predictions, we may use wider tubes.

However, the problem of interval regressors is that they are not as precise as regular regressors. There is some degree of vagueness in interval predictions. For this reason, following [15], we call them nondeterministic predictors.

To handle this type of predictions, we have to reformulate the alarms associated with a regressor. We need to interpret predictions that include, at the same time, alarms and non-alarms. Our proposal is to label these situations as *warnings*: something between alarms and non-alarms. The scores reported in Section 6 will illustrate the advantages of nondeterministic alarm functions.

Formally, we propose the following extension of (Eq. 3). Given an interval regressor h_{ND} , if τ is a threshold in \mathcal{Y} , we shall interpret the outputs of the regressor as follows

$$Alarm(h_{ND}(\mathbf{x})) = \begin{cases} \text{non-alarm} & h_{ND}(\mathbf{x}) \subset (-\infty, \tau] \\ \text{alarm} & h_{ND}(\mathbf{x}) \subset (\tau, +\infty) \\ \text{warning} & \text{otherwise.} \end{cases} \quad (7)$$

Additionally, from the classification point of view, given a nondeterministic regressor, h_{ND} , for a test set $S' = \{(\mathbf{x}'_1, y'_1), \dots, (\mathbf{x}'_m, y'_m)\}$, it is important to measure the proportion of test examples that fall outside the tube

$$out_tube(h_{ND}, S') = \frac{1}{m} \sum_{i=1}^m 1 - (y'_i \in h_{ND}(\mathbf{x}'_i)). \quad (8)$$

5 Time Series Alarms

As was explained in Section 2.1, an alarm system for coffee rust would be able to be used at any time. To simulate this capacity, given an incidence percentage y measured the first day of one month, we considered different values $t \in [30, 25, 20, 15, 10]$ for the number of days ahead of predictions. For each t we have the corresponding weather records. Thus, in the learning task, for each y we have a time series

$$\{(\mathbf{x}_t, y) : t \in [30, 25, 20, 15, 10]\}. \quad (9)$$

To evaluate the sequence of alarm alerts produced by an interval regressor h_{ND} in (Eq. 9), the idea is that if an alarming prediction occurs for some t , then the reaction would be to use the agrochemical fungicides; any subsequent notice of non-alarm would not be heard. On other hand truly non-alarming predictions for y would need a sequence of non-alarms for all t values. Formally, this point of view is captured by the following definition

$$Alarm(h_{ND}(\mathbf{x}_t)) = \begin{cases} \text{non-alarm} & \forall t, h_{ND}(\mathbf{x}_t) \subset (-\infty, \tau] \\ \text{alarm} & \exists t, h_{ND}(\mathbf{x}_t) \subset (\tau, +\infty) \\ \text{warning} & \text{otherwise.} \end{cases} \quad (10)$$

Table 1. Regression scores obtained (using cross-validation) for different values of the radius ϵ of the insensitive zone or *tube*. In rows, for each combination of fruit load (l) and spacing (s), we report the averages of *absolute error*, ϵ -insensitive loss, Δ_ϵ (Eq. 2), and *correlations*. The last rows shows the scores considering at the same time all types of plantations.

| | Score | $\epsilon = 0.1$ | $\epsilon = 1$ | $\epsilon = 2$ | $\epsilon = 3$ | $\epsilon = 4$ |
|--------------------|-----------------------|------------------|----------------|----------------|----------------|----------------|
| $l = 1$ $s = 1$ | <i>absolute error</i> | 6.53 | 6.11 | 5.86 | 5.75 | 5.72 |
| | Δ_ϵ | 6.53 | 6.02 | 5.62 | 5.23 | 4.79 |
| | <i>correlation</i> | 0.81 | 0.82 | 0.83 | 0.84 | 0.84 |
| | <i>out_tube</i> | 0.98 | 0.80 | 0.73 | 0.65 | 0.54 |
| $l = 1$ $s = 2$ | <i>absolute error</i> | 7.40 | 7.07 | 6.87 | 6.88 | 7.13 |
| | Δ_ϵ | 7.40 | 7.00 | 6.60 | 6.35 | 6.41 |
| | <i>correlation</i> | 0.82 | 0.82 | 0.82 | 0.82 | 0.82 |
| | <i>out_tube</i> | 1.00 | 0.86 | 0.73 | 0.63 | 0.61 |
| $l = 2$ $s = 1$ | <i>absolute error</i> | 6.45 | 6.33 | 6.20 | 6.14 | 6.29 |
| | Δ_ϵ | 6.45 | 6.27 | 5.96 | 5.71 | 5.40 |
| | <i>correlation</i> | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 |
| | <i>out_tube</i> | 0.98 | 0.88 | 0.76 | 0.66 | 0.55 |
| $l = 2$ $s = 2$ | <i>absolute error</i> | 6.56 | 6.13 | 5.84 | 5.73 | 5.80 |
| | Δ_ϵ | 6.56 | 6.05 | 5.59 | 5.16 | 4.90 |
| | <i>correlation</i> | 0.96 | 0.96 | 0.97 | 0.97 | 0.97 |
| | <i>out_tube</i> | 0.99 | 0.83 | 0.75 | 0.64 | 0.56 |
| <i>all</i> | <i>absolute error</i> | 6.74 | 6.41 | 6.19 | 6.12 | 6.23 |
| | Δ_ϵ | 6.73 | 6.34 | 5.94 | 5.61 | 5.38 |
| | <i>correlation</i> | 0.94 | 0.94 | 0.94 | 0.94 | 0.95 |
| | <i>out_tube</i> | 0.98 | 0.84 | 0.74 | 0.64 | 0.57 |

6 Experimental Results

In this section we report a number of experiments carried out to illustrate the role played by the width of intervals involved in alarm predictions. With the dataset introduced in Section 2.1, we used a 10-fold cross validation to estimate the scores reported in the following figure and tables. As was mentioned in the introduction, in all the experiments, the threshold used to discriminate alarms was $\tau = 4.5$.

The SVM regressors were learned using LibSVM [3], with an *rbf* kernel. The parameters C and σ were adjusted using an *internal grid search* in each training set. The ranges for this search were: $C \in [0.001, 0.01, 0.1, 1, 10, 100, 1000]$, and $\sigma \in [0.01, 0.1, 0.3, 0.5, 0.7]$. The search employed an internal 2-fold cross validation repeated 3 times; the aim being to optimize the average Δ_ϵ (Eq. 2).

First we compared the scores achieved from the point of view of regression for different values of ϵ and different plantation types. The results are gathered in Table 1.

We observe that correlations are quite different from the data of plantations with low ($l = 1$) and high ($l = 2$) fruit loads. The quality of regressors is worse in

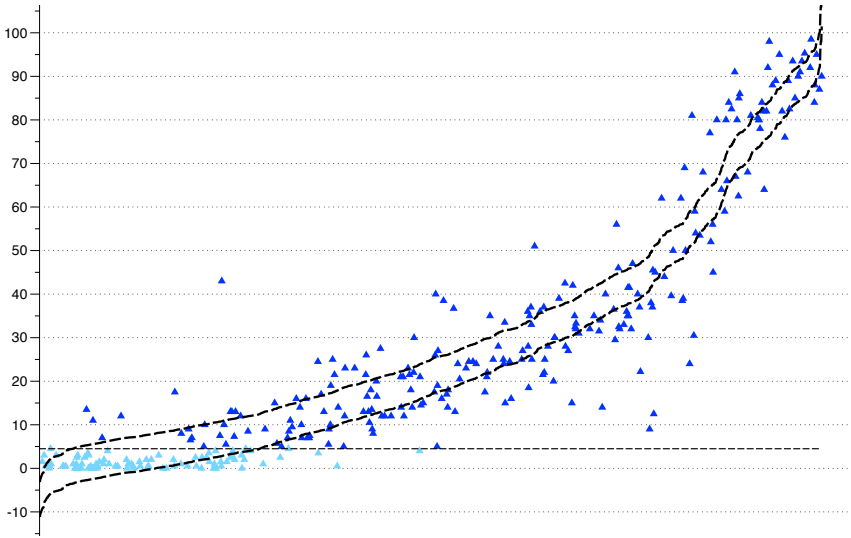


Fig. 1. True incidence percentages (\blacktriangle) and the predicted intervals by a regressor $h_{ND(4)}$. The horizontal axis represents the indexes of samples ordered according to their predictions. We only included predictions made one month in advance to make the figure more clear. The proportion of points outside the tube are similar if we vary the days ahead of predictions. The horizontal dashed line represents the threshold $\tau = 4.5$.

the case of low fruit load. However, the correlation obtained for the whole dataset is quite high, around 0.94. The value of the radius of the predicted interval, ϵ , has no influence on these results. But, of course, ϵ has a dramatic impact in the proportion of points outside the tube. Here, the results range from almost all to 0.57. Obviously, it is easier to include examples inside wider tubes.

In Figure 1 we represent graphically the predictions and true values. To make the figure more clear, we show only a subset of examples: predictions made one month ahead. We used the predictions of the interval regressor learned with $\epsilon = 4$, $h_{ND(4)}$; that is, a regressor whose predictions are intervals with a width of 8. We can appreciate that the errors are higher when predictions range from 30 to 50.

Time series. In Table 2 we report the results obtained by the alarm functions obtained for the time series described in Section 5. In this case, in cross-validations we took care that time series (Eq. 9) were never separated into train and test splits.

The table shows the confusion matrices obtained in cross-validations. For *deterministic* regression, the default value of the insensitive zone used was $\epsilon = 0.1$. In this case, of course, there are no doubtful classifications: no warnings appear in the corresponding columns of Table 2. Unfortunately, the consequence is that the number of errors is too high: 14 false non-alarms, and 16 false alarms.

Table 2. Confusion matrices obtained (using cross-validation over time series) for different values of the radius ϵ of the insensitive zone or *tube*. Columns represent true classes: alarm (a), non-alarm ($\neg a$). Rows report the occurrences of each possible prediction (Pre) (alarm, warning (w), non-alarm) for each combination of load (l) and spacing (s). The last rows shows the scores considering at the same time all types of plantations; that is, the sum of the corresponding confusion matrices.

| | | $\epsilon = 0.1$ | | $\epsilon = 1$ | | $\epsilon = 2$ | | $\epsilon = 3$ | | $\epsilon = 4$ | | |
|---------|----------|------------------|----------|----------------|----------|----------------|----------|----------------|----------|----------------|----------|-----|
| | | Pre | $\neg a$ | a | $\neg a$ | a | $\neg a$ | a | $\neg a$ | a | $\neg a$ | a |
| $l = 1$ | $\neg a$ | 18 | 6 | 15 | 3 | 12 | 0 | 3 | 0 | 1 | 0 | |
| $s = 1$ | w | 0 | 0 | 3 | 4 | 7 | 9 | 15 | 9 | 18 | 9 | |
| | a | 5 | 56 | 5 | 55 | 4 | 53 | 5 | 53 | 4 | 53 | |
| $l = 1$ | $\neg a$ | 18 | 6 | 12 | 2 | 6 | 0 | 1 | 0 | 1 | 0 | |
| $s = 2$ | w | 0 | 0 | 6 | 4 | 10 | 6 | 15 | 2 | 17 | 5 | |
| | a | 5 | 56 | 5 | 56 | 7 | 56 | 7 | 60 | 5 | 57 | |
| $l = 2$ | $\neg a$ | 16 | 0 | 14 | 0 | 11 | 0 | 3 | 0 | 2 | 0 | |
| $s = 1$ | w | 0 | 0 | 2 | 0 | 5 | 0 | 13 | 0 | 14 | 0 | |
| | a | 4 | 65 | 4 | 65 | 4 | 65 | 4 | 65 | 4 | 65 | |
| $l = 2$ | $\neg a$ | 17 | 2 | 14 | 0 | 5 | 0 | 3 | 0 | 1 | 0 | |
| $s = 2$ | w | 0 | 0 | 2 | 1 | 12 | 0 | 13 | 0 | 16 | 0 | |
| | a | 2 | 64 | 3 | 65 | 2 | 66 | 3 | 66 | 2 | 66 | |
| sum | $\neg a$ | 69 | 14 | 55 | 5 | 34 | 0 | 10 | 0 | 5 | 0 | |
| | w | 0 | 0 | 13 | 9 | 34 | 15 | 56 | 11 | 65 | 14 | |
| | a | 16 | 241 | 17 | 241 | 17 | 240 | 19 | 244 | 15 | 241 | |

If we use wider predicted intervals ($\epsilon \geq 1$), the number of errors decreases dramatically, but the price is that there is an increase in the number of warning predictions. Thus, for $\epsilon = 1$ the number of false non-alarms is only 5 with 22 warnings (6.5% of all cases). Let us remark that all these false non-alarms are due to plantations with low fruit load ($l = 1$), which is coherent with the results obtained for regression scores, see Table 1. With $\epsilon \geq 2$, the number of false non-alarms is zero, but the warnings rise to 14.4%, 19.7% and 23.2% respectively for $\epsilon = 2, 3, 4$.

7 Conclusion

We discussed the viability of an alarm system for coffee rust, the main coffee crop disease in the world. In this case, the aim is to apply the chemical prevention of the diseases only when necessary to achieve healthier products and reductions in cost and environmental impact. But we must be vigilant to avoid false non-alarms since they would conduct to not prevent an awful increase in the incidence of the disease.

The approach presented here proposes to handle predictions about continuous variables by regressors able to predict intervals rather than single points. They can be learned from regression learning tasks using the so-called ϵ -insensitive zone (ϵ is the radius of the predicted intervals). An optimal solution can be obtained by Regression Support Vector Machines.

The use of interval predictors allow us to distinguish a third type of situations placed between alarms and non-alarms. We called them warnings. Roughly speaking, we found that the confidence in non-alarm predictions is higher as ϵ increases, while it is quite stable for alarm predictions. Somehow, the alarm predictor becomes more prudent, but requires more frequently deeper analysis to decide what to do in uncertain (warning) situations.

A trade off between the number of non-alarms and warnings would lead to a useful alarm system for the coffee rust. If we want to search for an optimal value for ϵ , we must consider the important economic and environmental aspects involved in coffee growing.

Finally, it is worth noting here that the cost of implementing the alarm systems presented in this paper is very low. The only requirement is a cheap weather station able to register the data described in Section [2.1](#).

Acknowledgements

The authors are grateful to the Brazilian Fundação Pró Café for providing the data used in this paper.

References

1. Alonso, J., del Coz, J.J., Díez, J., Luaces, O., Bahamonde, A.: Learning to predict one or more ranks in ordinal regression tasks. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 39–54. Springer, Heidelberg (2008)
2. Bartlett, P., Wegkamp, M.: Classification with a reject option using a hinge loss. *Journal of Machine Learning Research* 9, 1823–1840 (2008)
3. 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>
4. Chow, C.: On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory* 16(1), 41–46 (1970)
5. del Coz, J.J., Díez, J., Bahamonde, A.: Learning nondeterministic classifiers. *Journal of Machine Learning Research* 10, 2273–2293 (2009)
6. Japiassú, L., Garcia, A., Miguel, A., Carvalho, C., Ferreira, R., Padilha, L., Matiello, J.: Influência da carga pendente, do espaçamento e de fatores climáticos no desenvolvimento da ferrugem do cafeeiro. In: Simpósio de Pesquisa dos Cafés do Brasil, Águas de Lindóia, SP, Brasil (2007)
7. Meira, C., Rodrigues, L., de Moraes, S.: Análise da epidemia da ferrugem do cafeeiro com árvore de decisão. *Tropical Plant Pathology* 33(2), 114–124 (2008)
8. Meira, C., Rodrigues, L., de Moraes, S.: Modelos de alerta para o controle da ferrugem-do-cafeeiro em lavouras com alta carga pendente. *Pesq. agropec. bras* 44(3), 233–242 (2009)

Prediction of Web Goodput Using Nonlinear Autoregressive Models^{*}

Maciej Drwal and Leszek Borzemski

Institute of Informatics
Wrocław University of Technology
Wrocław, Poland
{maciej.drwal,leszek.borzemski}@pwr.wroc.pl

Abstract. The performance prediction is a key part of the modern network traffic engineering. In this paper we present the application of nonlinear autoregressive modeling to the prediction of goodput level in web transactions. We propose the two-stage approach, with clustering step on historical data, prior to classification, to determine the most appropriate traffic intensity levels. Our study is based on the data collected by the *MWING* system, an ensemble of web performance measurement agents, and cover over a year of continuous observations of a group of HTTP servers.

Keywords: traffic engineering, machine learning, Internet applications.

1 Introduction

Many network applications benefit greatly from the utilization of reliable load forecasting. Examples of such application include content distribution systems, grid computing schedulers and load balancing systems in specialized overlay networks.

We have developed a specialized measurement system for the purpose of web traffic measurements. The *MWING* (which stands for *multi-agent web-ping*, [3]) is a distributed measurement framework, designed as a platform for the continuous probing of the end-to-end HTTP transaction performance. Deploying a dedicated measurement system in the Internet, instead of using already existing platform (like PlanetLab, [15]), allowed us to obtain possibly reliable real-world traffic data. In addition, *MWING* system potentially gives the maximum flexibility and control over the examination schemes, and is capable of carrying out other experiments.

In our research, we are interested in the prediction of the transmission rate as observed in a typical web server communication. We consider the estimation of goodput — the number of useful bytes transmitted in a time unit. Our aim is to find a way to predict the level of goodput on a given path, with the help of

^{*} This work was partially supported by the Polish Ministry of Science and Higher Education under Grant No. N516 032 31/3359 (2006—2009).

measurement system, similar in capabilities to *MWING*. The expected goodput gives the most practical information about the network performance, perceived by the user. However, due to its nature, the forecasting of goodput is generally a difficult problem.

2 Related Work

The predictability of the end-to-end throughput has been already studied from many different aspects. Various time series based and formula based approaches have been used for the performance forecasting (e.g. [16], [8]). For summary of the study of IP traffic nature (self-similarity, burstiness) see [1] and [14]. Recently, the methods from artificial intelligence and machine learning are more willingly applied to the prediction of Internet traffic characteristics (e.g. [10], [13]).

However, many of these ideas are still pending for a rigorous verification in the live experiments, performed in evolving Internet.

3 Feature Selection

The *MWING* system is designed to provide the estimates of throughput and goodput between a selected set of client machines and HTTP servers in the Internet. A single probe from a client gives the measured time periods of the following communication stages:

- **Dns** — time to resolve address via DNS lookup
- **Con** — time to establish TCP connection,
- **First** — time elapsed between sending HTTP GET request, and receiving the first response packet
- **Last** — remaining time of the HTTP transaction.

Additionally, in-between delays are measured, denoted as **D2S** (between the end of **Dns** and sending first TCP SYN packet), and **A2G** (between sending TCP ACK packet and sending GET request). The total measured time is $T = \text{Dns} + \text{D2S} + \text{Con} + \text{A2G} + \text{First} + \text{Last}$.

The goodput estimate is $\text{Gpt} = S/T$, where S denotes the transmitted resource size expressed in bytes. These S bytes include only the essential information we want to fetch from the server (e.g. hypertext file, video object, etc.), no matter how many additional bytes are needed to transmit it (for error correction, protocol stack information, etc.). The goodput can be thought of as an application level throughput.

MWING performs the measurements continuously in 30 minutes intervals. Consequently, we consider only the effects discernible on at least 1 hour spans. All the measurements were performed simultaneously from four client machines (called measurement agents). Three of them were located in the university networks in Poland: Wroclaw (**WRO**), Gdansk (**GDA**), Gliwice (**GLI**). The fourth one was located in Las Vegas, USA (**LAS**).

Table 1. An example correlation matrix, from one month data (957 measurements on GDA—curl.nedmirror.nl path). Significant predictor-dependent variable pairs are highlighted. **LogGpt** is better correlated with measured values. Day of week (**DoW**) in this dataset is less relevant (possibly the coverage is too low to exhibit weekly trends).

| | Dns | D2S | Con | A2G | First | Last | Hour | DoW | LogGpt | Gpt |
|--------|---------------|--------|---------------|--------|---------------|---------------|---------------|--------|---------------|---------------|
| Dns | 1.000 | 0.164 | 0.046 | 0.013 | 0.060 | 0.024 | 0.090 | -0.070 | -0.604 | -0.523 |
| D2S | 0.165 | 1.000 | -0.007 | 0.004 | 0.037 | 0.057 | 0.048 | -0.051 | -0.164 | -0.177 |
| Con | 0.047 | -0.007 | 1.000 | 0.105 | 0.124 | 0.033 | 0.008 | -0.018 | -0.409 | -0.261 |
| A2G | 0.013 | 0.004 | 0.105 | 1.000 | 0.060 | 0.156 | 0.029 | -0.042 | -0.139 | -0.115 |
| First | 0.060 | 0.037 | 0.124 | 0.061 | 1.000 | 0.115 | 0.053 | -0.041 | -0.421 | -0.335 |
| Last | 0.023 | 0.057 | 0.033 | 0.155 | 0.115 | 1.000 | 0.127 | -0.031 | -0.501 | -0.462 |
| Hour | 0.090 | 0.048 | 0.008 | 0.029 | 0.0534 | 0.127 | 1.000 | -0.016 | -0.192 | -0.249 |
| DoW | -0.070 | -0.051 | -0.018 | -0.042 | -0.041 | -0.031 | -0.016 | 1.000 | 0.082 | 0.085 |
| LogGpt | -0.604 | -0.163 | -0.409 | -0.139 | -0.421 | -0.501 | -0.192 | 0.082 | 1.000 | 0.940 |
| Gpt | -0.522 | -0.177 | -0.261 | -0.113 | -0.335 | -0.462 | -0.249 | 0.086 | 0.940 | 1.000 |

Since the performance of a path could be affected by many unknown factors [8], it is reasonable to make use of several different predictors. Some short-term factors, which we do not measure directly, are caused, for example, by the cross-traffic on the parts of the path from client to server, the instantaneous server load, and changes in intermediate hop directions due to the routing algorithms. The long-term factors are, for example, major changes in routing tables or changes of the server’s hardware or software.

Consequently, the goodput level prediction procedure will make use of the historical measurements to grasp the long-term factors. Because the older the observations we take into account, the more we have to “average” over the relevant information, we reduce the dataset taken into account to a shifting-window of a fixed size. The decision needs to be made upon the short-term influences as well, thus we assume that we make a prediction for the instant of time directly after a performed measurement. This approach leads to the use of time series analysis. We propose the adaptation of autoregressive moving average models [4], as discussed in detail in Section 5.

In the first step of our analysis we considered correlation matrices estimated for various measurement periods. We found the most significant predictors for **Gpt** and **LogGpt** random variables, as illustrated in Table 1. It is evident that in most cases the coefficients for **LogGpt** are prevailing, since the partial times show stronger linear dependencies, compared to **Gpt**. There is, as expected, strong negative correlation with **Dns**, **Con** and **First** values (the goodput is inversely proportional to the sum of these time periods). The **Last** value describes the major part of client-server transaction time, however in case of long transactions, the *MWING*-type measurement of this value cannot be used for on-line prediction. We also leave out the **D2S** and **A2G**, to reduce the number of features in the model.

¹ The logarithm of base 10.

We have performed tests for statistical significance of the periodic exogenous factors in [6], i.e. time of the day (**Hour**), and day of the week (**DoW**). As expected, in the majority of cases, the influence of **Hour** is evident, and this is supported by our results. Taking **DoW** is reasonable in some cases for longer training sets. We used both standard one-way F-statistic ANOVA test, as well as rank based Kruskal-Wallis test. These test may reveal both linear and nonlinear dependencies. Assuming the level of significance $\alpha = 0.05$, the influence of the day of the week was observed as follows: 86% (**LAS**), 85% (**WRO**), 64% (**GDA**), 46% (**GLI**)². Similarly, the influence of the local hour is proven to be significant; our test results were: 56% (**LAS**), 87% (**WRO**), 62% (**GDA**), 50% (**GLI**).

The goodput time series are nonlinear, which is easy to show, even for short measurement periods, e.g. with the use of BDS test [5]. In essence, such time series should be considered as a composition of deterministic (chaotic) and stochastic processes, see Fig. 1.

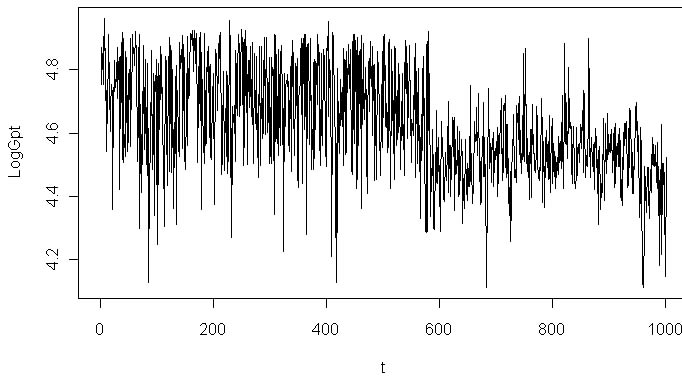


Fig. 1. An example of 1000 subsequent measurements of **LogGpt** on a single path (lasting for nearly 2 months). There is a significant drop slightly after the half.

In order to justify the use of autoregressive part in our models, we have analyzed the autocorrelation of goodput time series. The estimated values were usually high only for very short lags, see Fig. 2. This explains the general low predictability of the goodput. Nevertheless, the time-local information on the trend can be very useful, as we cannot tell in advance how correlated the series will be.

In summary, the general model to learn from the *MWING* datasets is:

$$\text{LogGpt}_n \sim f(\text{LogGpt}_{n-1}, \text{LogGpt}_{n-2}, \dots; \text{Dns}, \text{Con}, \text{First}, \text{Hour}, \text{DoW}) \quad (1)$$

where **Dns**, **Con** and **First** are continuous variables, and **Hour** (0–23) and **DoW** (1–7) denote the local time instant of the measurement. The last two variables are considered as the exogenous inputs. Each input is normalized onto the $[0, 1]$

² The ratio of significant cases, among all tests performed for a given client.

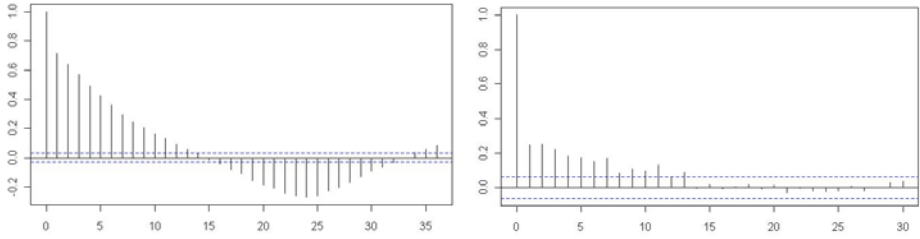


Fig. 2. The autocorrelation estimate for an excerpt of observations (one month), as the function of time lag. On the left: a predictable path. On the right: a path with lower predictability (autocorrelation quickly drops to 0).

interval, via the formula: $z_i = (x_i - \min x_i) / (\max x_i - \min x_i)$. For class-based prediction, only a discrete set of values of **LogGpt** are allowed, each representing certain traffic level.

The function f can be any appropriate nonlinear model, as discussed further in Section 5. Such models may make use of higher order predictors, or nonlinear transformations of predictors. In the time series modelling terminology, these models can be classified as a nonlinear autoregressive exogenous models (NARX).

4 Performance Level Differentiation via Clustering

We have analyzed both classification and regression problems, which arise in the web traffic engineering. The prediction of exact goodput value is a difficult task, however for most applications we are interested only in its average level. We utilize the two-stage data mining approach, in order to determine (and be able to update later) the appropriate traffic intensity levels.

On a fixed client–server path we usually observe cumulation of goodput around typical values. Because of the various slowing factors, appearing at random, the goodput distributions are generally not unimodal (see example on Fig. 3). In fact, the probability densities of **Gpt** (and **LogGpt**) for an end-to-end connection would be best modelled as a composition of a small number of shifted bell-shaped curves, as a mixture model:

$$f(x) = \sum_{n=1}^K a_n f_n(x; \Theta_n) \tag{2}$$

where $\sum_{n=1}^K a_n = 1$. Simplifying the description, each component could count for one setup in the routing tables of the network core; the longer we observe the link, the more such components may appear.

If we assume the gaussian components $f_n(x; \Theta_n)$, we can make use of maximum likelihood principle for estimating parameters of model Θ_n (assuming the number of classes K). The expectation-maximization algorithm (EM) is a well

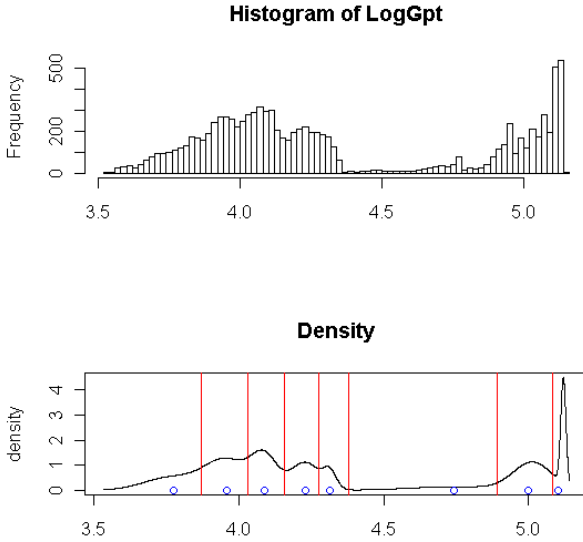


Fig. 3. An example result of EM clustering on a single client-server path. The histogram is used to estimate the gaussian mixture of 8 components. The dots denote the locations of means μ_i , while vertical lines separate performance classes in points x_i .

known tool, useful for this kind of problems [12]. This approach has been already successfully used in the Internet traffic modelling, see [11].

Instead of assuming the number of classes in advance, we can perform the EM based clustering, to find the best number of components K (in a desired range). This technique makes use of the Bayesian Information Criterion [9], to evaluate the quality of model for different K . The subsequent runs of EM algorithm result in models for which the maximized value of the likelihood function is taken for comparison. The BIC criterion is estimated as $RSS/\sigma_e^2 + k \ln(n)$, where RSS is the residual sum of squares, σ_e is the error variance (we assume normal distribution of errors), $k = 2$ is the number of free parameters, and n is the sample size. We repeat the whole process for different K , and select the model with lowest BIC, minimizing the average log-likelihood.

Knowing the components $f_n(x; \Theta_n)$ we calculate the decision boundaries, i.e. the points x_i separating classes: $\forall_{i=1, \dots, K-1} x_i : f_i(x_i; \Theta_i) = f_{i+1}(x_i; \Theta_{i+1})$.

The advantage of Gaussian components is that the boundary points can be found easily, in the same way as in the linear discriminant analysis. Taking the logarithms of two components, the problem simply reduces to the solution of quadric equation (if variances happen to be the same, this reduces to linear equation). Let (μ_1, σ_1) and (μ_2, σ_2) be two components. The boundary points are obtained by solving:

$$(\sigma_1^2 - \sigma_2^2)x^2 + 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2)x + \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 - 2(\sigma_1\sigma_2)^2 \ln(\sigma_1/\sigma_2) = 0$$

with respect to x . Now with each of the K intervals:

$$[0, x_1), [x_1, x_2), \dots, [x_{K-1}, \infty)$$

we associate a performance (average goodput level) class. Note that their sizes can vary significantly, see Fig. 3. Moreover, for a given server, one clustering stays valid only for a limited time.

5 Predictive Analysis

In order to obtain a good predictive model, which is both accurate and flexible for the on-line learning, we have tested several state of the art machine learning algorithms. The full presentation of our research is beyond the scope of this text. Here we present two examples of NARX classifiers, which were among the top performers in the general settings: the autoregressive exogenous model with neural network classifier (NNET), and with k -nearest neighbor classifier (KNN).

Our approach to the time series forecasting is based on the adapting off-the-shelf statistical classifiers into the autoregressive models. This can be thought of as the on-line learning paradigm, since we would like to use the predictor continuously in time, and expect it to adapt to the changing conditions. Such classifier tries to reconstruct the dynamics of the underlying process.

Virtually any classifier f can be adopted to the Model 1, possibly with some special tweaks. A NARX classifier is provided with a shifting window of a fixed size of w last observations of the feature vector. This vector includes both endogenous inputs (**Dns**, **Con**, **First**) and exogenous (**Hour**, **DoW**). Additionally, it includes the autoregressive part, i.e. the previous l values of **LogGpt** (l is the lag parameter). After training the classifier with a set of w observations, we make one prediction, the value of LogGpt_{w+1} . For all subsequent predictions, we drop the oldest measured **LogGpt** from the training set (along with the oldest training vector), and include the predicted one.

The autoregressive part contributes to the detection of local trends in the time series, while the remaining part allows for seeking the best probabilistic decision, given a feature vector.

The NNET model makes use of 2-layer feedforward backpropagation neural network for the classification. The hidden layer typically contains 7–12 sigmoid units. The decision is made by rounding the output to the nearest class.

The KNN model uses nonlinear transformation of the input space. Let $r = \text{Dns} + \text{Con} + \text{First}$. It uses the terms of degree 3 polynomial of r as endogenous inputs. Given an input vector, the decision is made by selecting the class, for which the majority of k nearest training vectors belong. Such classifier tries to approximate the optimal (Bayesian) decision, $i^* = \arg \max P(i|x)$, by relaxing the conditional probability within a small neighborhood of the feature vector instance x 7. This classifier is considered nonlinear, as its resulting decision boundaries can adjust to any shape.

For the purpose of model performance comparison we use the lag parameter $l = 1$, and compare different shifting-window sizes w : 1 day, 3 days, 1 week,

2 weeks and 3 weeks. We have tested the models against a collection of goodput measurements, taken in between 2008.4.24 — 2009.7.5 by *MWING* system. The datasets contained missing values, due to the temporary server or network malfunctions. Some of the observed servers were subject to higher traffic than others and, in result, some of them provided a more challenging prediction problem than others. We did not take into consideration situations with very low variance traffic, for which the number of observations in different clusters is highly unequal (for example over 90% of training cases belonged to one cluster). In such cases the best prediction degrades to a priori probability decision.

Figure 4 shows the classification rate of the both models for 25 servers. For each server the EM clustering was run prior to the classification (there were 4—5 clusters at average). The clustering stage was repeated after every 5 full window shifts, updating the traffic levels' boundaries. Figure 5 presents an example single server case: the boundaries are updated 6 times, each update accounts for one week. This shows how the overall traffic shaping can vary in time.

For a small group of considered servers a very accurate prediction (over 90% correct) was achieved. For the majority of them, with correctly set up class boundaries, it was possible to achieve 75—80% classification rate.

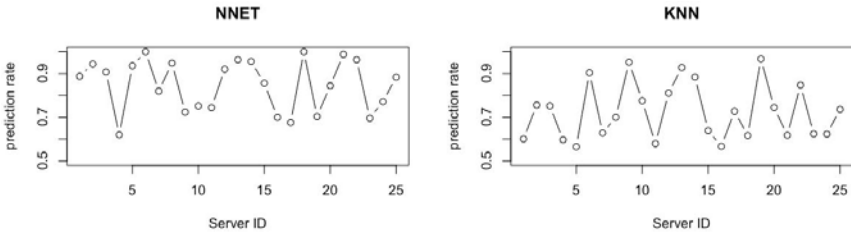


Fig. 4. The ratio of correctly classified goodput levels for WRO client and 25 web servers. Comparison of performance of neural network and nearest neighbor models for $w = 672$ (2 weeks window).

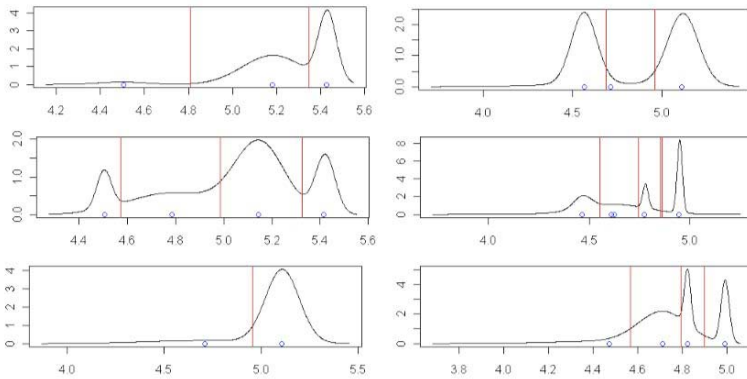


Fig. 5. Subsequent clustering updates for one observed end-to-end network path

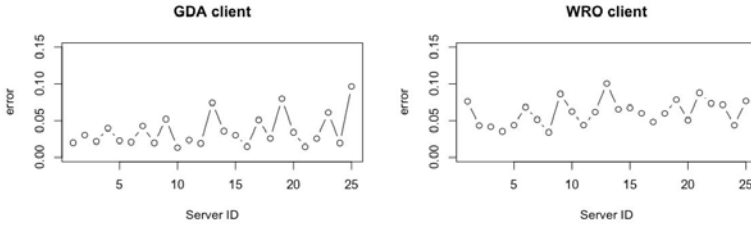


Fig. 6. Mean regression error of **LogGpt** using neural network based predictor. The GDA client measurements gave better results in many datasets.

For comparison, we also considered reverse approach, i.e. solving analogous regression problem (prediction of exact value of **LogGpt**), followed by discretization into previously clustered classes. Using exactly the same prediction scheme, the average regression error was around 5–7% (depending on client/server dataset, see Fig. 6).

Generally, the NNET model performed substantially better, however for some datasets the performance was comparable. A big advantage of the KNN model is that its implementations can be very fast. For the reasonable sizes of the shifting window the decision making is instant. This allows for the on-line real-time parameter tuning, holding simultaneously a group of similar classifiers. In both cases the automated selection of the best classifier can be applied, given enough computational resources. Even with delayed updates, the models stay fairly accurate for many weeks.

6 Conclusions

Next generation networks, as we believe, would benefit from the use of performance prediction techniques. In particular, there is a strong interest among network operators and providers in obtaining the full control over the quality of Internet services. Predictive analysis can be also very useful for designing customized network solutions.

The nonlinear autoregressive models with exogenous inputs appear to be among the methods of choice for this purpose. These models combine the advantages of the time series based prediction techniques with the static machine learning inference for cumulated observations. Using neural network based predictors, adapted to our on-line learning scheme, we have achieved even over 90% prediction rate on several web servers. With the combination of gaussian mixture clustering, the presented two-stage methodology allows for the fully automatic recognition of the web traffic intensity in the Internet.

References

1. Abry, P., Baraniuk, R., Flandrin, P., Riedi, R., Veitch, D.: The multiscale nature of network traffic: Discovery, analysis, and modelling. *IEEE Signal Processing Magazine* 19(3), 28–46 (2002)

2. Baccelli, F., McDonald, D.R.: A stochastic model for the throughput of non-persistent TCP flows. *Performance Evaluation* 65(6–7), 512–530 (2008)
3. Borzemski, L., Cichocki, L., Kliber, M.: Architecture of Multiagent Internet Measurement System MWING Release 2. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2009*. LNCS, vol. 5559, pp. 410–419. Springer, Heidelberg (2009)
4. Box, G., Jenkins, G.M., Reinsel, G.C.: *Time Series Analysis: Forecasting and Control*, 3rd edn. Prentice-Hall, Upper Saddle River (1994)
5. Brock, W.A., Dechert, W.D., Scheinkman, J.A., LeBaron, B.: A Test for Independence Based on the Correlation Dimension. *Econometric Reviews* 15(3), 197–235 (1996)
6. Drwal, M., Borzemski, L.: Statistical Analysis of Active Web Performance Measurements. In: 6th Working Conference HET-NETs 2010, pp. 247–258 (2010)
7. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning Data Mining, Inference, and Prediction*, 2nd edn. Springer, Heidelberg (2009)
8. He, Q., Dovrolis, C., Ammar, M.: On the Predictability of Large Transfer TCP Throughput. *Computer Networks* 51(14), 3959–3977 (2007)
9. Keribin, C.: Consistent estimate of the order of mixture models. *Comptes Rendus de l'Academie des Sciences Series I Mathematics* 326(2), 243–248 (1998)
10. Kim, H., Claffy, K.C., Fomenkov, M., Barman, D., Faloutsos, M., Lee, K.Y.: Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices. In: ACM CoNEXT conference, article no. 11. ACM, New York (2008)
11. Liu, Z., Almhana, J., Choulakian, V., McGorman, R.: Online EM Algorithm for Mixture with Application to Internet Traffic Modeling. *Computational Statistics and Data Analysis* 50(4), 1052–1071 (2006)
12. McLachlan, G.J., Krishnan, T.: *The EM Algorithm and Extensions*. John Wiley & Sons Inc., Chichester (1997)
13. Mirza, M., Sommers, J., Barford, P., Zhu, X.: A machine Learning Approach to TCP Throughput Prediction. In: ACM SIGMETRICS 2007 Conference, pp. 97–108 (2007)
14. Park, K., Willinger, W.: *Self-similar Network Traffic and Performance Evaluation*, 1st edn. Wiley-Interscience, New York (2000)
15. Peterson, L., Bavier, A., Fiuczynski, M., Muir, S.: Experiences Building PlanetLab. In: 7th symposium on Operating systems design and implementation OSDI 2006, pp. 351–366 (2006)
16. Sang, A., Li, S.-q.: A predictability analysis of network traffic. *Computer Networks* 39, 329–345 (2002)

Domain Driven Data Mining for Unavailability Estimation of Electrical Power Grids

Paulo J.L. Adeodato^{1,2}, Petrônio L. Braga², Adrian L. Arnaud¹,
Germano C. Vasconcelos^{1,2}, Frederico Guedes³, Hélio B. Menezes³,
and Giorgio O. Limeira³

¹ NeuroTech Ltd.,

Av. Cais do Apolo, 222 / 8º andar, 50030-905, Recife-PE, Brazil

² Center for Informatics, Federal University of Pernambuco,

Av. Professor Luís Freire s/n, Cidade Universitária, 50740-540, Recife-PE, Brazil

³ Companhia Hidrelétrica do São Francisco - CHESF,

St. Delmiro Gouveia, 333 – Bongi, 50761-901, Recife-PE, Brazil

{Paulo, Adrian, Germano}@neurotech.com.br,

{pjla, plb, gcv}@cin.ufpe.br, {fred, helio, giorgio1}@chesf.gov.br

Abstract. In Brazil, power generating, transmitting and distributing companies operating in the regulated market are paid for their equipment availability. In case of system unavailability, the companies are financially penalized, more severely, on unplanned interruptions. This work presents a domain driven data mining approach for estimating the risk of systems' unavailability based on their component equipments historical data, within one of the biggest Brazilian electric sector companies. Traditional statistical estimators are combined with the concepts of Recency, Frequency and Impact (RFI) for producing variables containing behavioral information finely tuned to the application domain. The unavailability costs are embedded in the problem modeling strategy. Logistic regression models bagged via their median score achieved Max_KS=0.341 and AUC_ROC=0.699 on the out-of-time data sample. This performance is much higher than the previous approaches attempted within the company. The system has been put in operation and will be monitored for the performance re-assessment and maintenance re-planning.

Keywords: Electrical power grid unavailability, Equipment unavailability penalties, Domain driven data mining, Model ensembles, Logistic regression.

1 Introduction

In the 1990s, most of the Brazilian power companies went private and started operating, under concession from the government, regulated by the National Agency of Electrical Energy (ANEEL = Agência Nacional de Energia Elétrica) and inspected by the National System Operator (ONS = Operador Nacional do Sistema). The companies operating in this regulated market are paid for the service they provide and are penalized for system unavailability at the Operational Function (FUNOP = FUNção OPERacional) level [1]. Each unavailability penalty depends on the value of the

FUNOP asset, its characteristics, the duration of the power interruption and, mainly, if the interruption had been planned or not; an unplanned unavailability costs roughly 20 times more than a planned one of the same duration [1].

The reliability of electrical power grids is already very high and under continuous improvement. Each FUNOP is composed of several equipments which implement an operational function in power generation, transmission or distribution. For preserving this high reliability profile, strict maintenance plans are periodically conducted on these equipments, with particular features for each family of equipments.

In general, the maintenance plan is made according mainly to the equipment manufacturer's recommendations. That takes into account the electrical load, the temperature and other aspects to define the periodicity, the procedures and parameter monitoring and adjustments. The equipment manufacturers have carried out series of trials within their plants and also collect data from their costumers' installations and apply statistical methods for defining their maintenance recommendations.

However, there are many other factors interfering in the system reliability in different power grids such as the quality of the repairmen's labor, ways of loading the system *etc.* There is also a major aspect to be considered; as the system's quality improves, less data about risky conditions are produced. Therefore, the better the system becomes, the less data about faults will be available for statistical modeling of risky conditions. Fortunately, more data from monitoring operation in normal conditions are being collected and will be available for future modeling.

Instead of using the traditional statistical modeling, this paper introduces an approach based on behavioral data. That may seem odd if one thinks of the system operating in a stable regime, under a constant fault rate. However, as the faults are very rare events, it is not possible to assure a constant fault rate and the adherence in the hypothesis test always gives at least a small difference; behavioral consolidation of data may capture variations which are important for risk estimation. The results presented here support this idea.

This paper is organized in six more sections. Section 2 characterizes the unavailability problem faced by CHESF (Companhia Hidro Elétrica do São Francisco) with the data structure available and the integration and transformation needed. Section 3 shows the modeling of the problem as a binary decision based on the maintenance plan, the creation of behavioral variables and the selection of the most relevant variables for modeling. Section 4 describes the knowledge extraction process via a bagged ensemble of logistic regression models. Section 5 presents and interprets the results achieved on a statistically independent data set. Section 6 summarizes the important contributions, the limitations of the approach and future work to be done to broaden the research.

2 Problem Characterization

CHESF (Companhia Hidro Elétrica do São Francisco) is one of the biggest Power generating company in Brazil producing 10,618 MW in 14 hydroelectric power plants and 1 thermoelectric. It also transmits this energy along an 18-thousand km long power grid [2]. Its annual revenue has reached R\$ 5.64 billion (= US\$ 3.15 billion) in 2008, with a net profit of R\$ 1.43 billion. Unfortunately, the revenue losses caused by penalties for unavailabilities still remain undisclosed.

CHESF's power grid has 462 FUNOPs of 7 different families with an average of 39 equipments in a total of 17.8 thousand equipments with an average age of approximately 19 years of operation.

The seven different FUNOP families are: transmission lines, power transformers, reactors, capacitor banks, synchronous compensators, static compensators and isolated cables.

Just before being put in operation, the equipments and FUNOPs are registered in the Asset Management System (SIGA = Sistema de Gerenciamento de Ativos). After becoming operational, the equipments have all their maintenances, planned or not, recorded in the same system (SIGA).

Each unavailability, no matter the cause, is recorded in the accountability system within the Asset Management System (SIGA). Unavailabilities that occurred before of January 1, 2008 were recorded in the system (DISPON) which had no direct link to the SIGA system.

These two data sources hosted in two different systems with relational databases needed to be integrated in a single data mart because they are the basis for the unavailability risk estimation system to be developed.

The difference in granularity between the DISPON and SIGA databases and the consequent lack of a unique key together with the legacy systems turned this database integration into a non-trivial task.

Asset registration and their maintenance records have been integrated in the SIGA system for the last two years but there were several adjustments in data imported from legacy systems for previous periods in a much longer history.

The most important difficulty faced however was the integration with the DISPON system where each unavailability recorded had not been linked to an equipment maintenance action. Furthermore, DISPON had been abandoned without any data importation to the current SIGA installed only 2 years ago. So, the unavailability data were dumped from the legacy database (DISPON) and were joined to the current SIGA database to form the complete unavailability database. These integration steps alone took around 60% of the project duration, having required a lot of interactions with the IT management and electrical engineers at CHESF.

The purpose of this work is to estimate the risk of occurring unavailabilities in the FUNOPs which compose the power grid at each moment. At this point, it is important to emphasize a trick made to turn the risk assessment problem into a binary decision problem for data mining. Considering that unavailabilities caused by planned maintenances are negligible in cost compared to unplanned ones (only 1:20 ratio), and that maintenance actions reset the operational status of the system to optimal, the temporal sequence of planned maintenance actions defines a frame of time intervals where the presence or absence of unplanned unavailabilities characterize the binary target for the supervised training process. This binary target definition approach will be explained in the next section, along with the creation of behavioral information.

3 Data Transformation

The variables present in the integrated database in a relational architecture needed to be transformed into more meaningful variables concerning the binary decision problem

characterized for modeling the unavailability risk assessment problem. This section explains the proposed random variables that produce the most adequate mapping from the original input space to the data mart variables. It also presents how the binary decision target was defined.

3.1 Variable Creation

Behavioral data are widely used in behavior scoring for credit risk assessment [3] and other business applications. In that domain, in general, it consists of the RFM (Recency, Frequency and Monetary value) variables' creation approach [4]. For systems' faults at CHESF, the approach was adapted to capture the relevant sequential features implicit in each event within the FUNOP related to recency, frequency and impact along time for faults and errors (RFI approach). In this approach, the impact is measured by the duration, cost and other features related to each system component / event. This is a very important basis for systematic and automatic creation of behavioral variables, considering several time spans. Other variables inherent to the FUNOPs and related to their complexity were created, such as the amount of equipments, the families of equipments and the entropy of the equipment distribution within the FUNOP. This is a Domain Driven Data Mining approach [5] of embedding the expert's knowledge from the electrical engineering field into the decision support system. The RFI approach can be generalized to model rare events in several application domains where the impact is captured by several different metrics (to be published elsewhere).

Another important aspect is that, due to the very small amount of faults per equipment, their rate of faults is defined at the equipment family level. Several "ratio" variables were created for measuring differences from a FUNOP to the population. So, the ratio of the average rate of faults per family of equipments within a FUNOP and the average in the whole grid form an important set of variables. At this point, it is important to highlight that, in general, equipments are not replaced or swapped in the power grid; they are simply maintained.

3.2 Proposed Model and Label Definition

Considering the scarcity of data about system faults and the consequent high imprecision of the estimated distributions and fault rates, the approach adopted here was to convert a classical statistical problem into a data mining problem with the advantage of reducing the amount of limiting assumptions in the modeling process.

In this approach, the temporal sequence of planned maintenance actions defines a frame of time intervals used for modeling and labeling the system condition. The label is defined as "*bad*" if there is at least one unplanned unavailability within that time interval and, "*good*" otherwise. This characterizes the binary target needed for the supervised training process of the decision support system [6].

The set of all planned maintenances defines a sequence of time intervals, each of which possess a binary label and takes into account all the past history of the FUNOP and its components (behavior), as illustrated in Fig. 1.

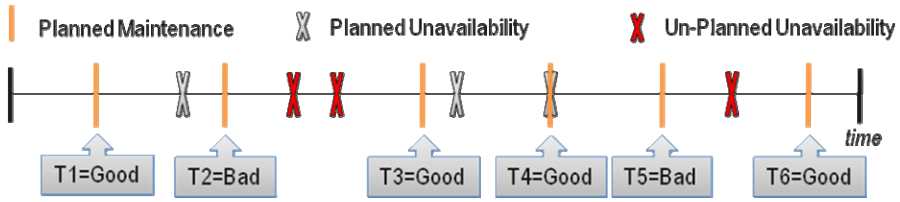


Fig. 1. Planned maintenances define a sequence of time intervals for modeling the problem as a binary decision and labeling the target

An approximation has been made in the approach depicted above, considering the negligible cost of the planned unavailabilities compared to the un-planned ones (1:20 ratio) and the fact that planned unavailabilities may be produced during a planned maintenance itself. Therefore, planned unavailabilities were discarded from the training data for the modeling process. No other constraint has been made concerning data distribution types or their parameters, different from the statistical approaches.

The goal of this modeling approach is to take preventive actions whenever a 'bad' prediction is made within a time interval. Despite not being in the long term maintenance plan, this short term planned maintenance action produces either negligible penalty (1:20 of the fault unavailability penalty) or no penalty at all (several preventive maintenance actions do not cause unavailability).

3.3 Variables Selection

As the process of systematic creation of behavioral variables makes it very easy to automatically produce new variables, variable selection is needed to preserve only the most meaningful and discriminative variables. The selection process was based on an approach for maximizing the information gain of the input variables in relation to the binary target and, simultaneously, minimizing the similarity (redundancy) among the input variables selected, measured by appropriate metrics, in a univariate fashion.

As all input variables were numerical and the target binary, the Max_KS (Kolmogorov-Smirnov) metric [7] was used for ranking the variables by their univariate discriminative importance. The redundancy among input variables was measured by linear correlation. The input variables with correlation higher than 0.9 with other variables of higher Max_KS were discarded from the model. Following this approach, only 30 among over 900 input variables were preserved. Table 1 lists the top five most relevant variables selected with their information gain measured in terms of Max_KS and AUC_ROC (Area under the ROC Curve) [8], to be explained in Sub-section 5.1.

It is clear that unplanned unavailability along the last two years of operation is the most relevant aspect for estimating the risk of unavailability before the next planned maintenance. It is interesting that the equipments' age appear only in 22nd place in the ranking with Max_KS=0.09 and AUC_ROC=0.48, suggesting that the system fault rate is indeed at the flat part of its curve.

Table 1. Five univariately most relevant variables selected in terms of Max_KS

| Variables | Max_KS | AUC_ROC |
|--------------------------------------------------------|--------|---------|
| Hours of UnPlanned Unavailability in Last 24 Months | 0.30 | 0.68 |
| Quantity of UnPlanned Unavailability in Last 24 Months | 0.28 | 0.69 |
| Hours of UnPlanned Unavailability in Last 12 Months | 0.27 | 0.66 |
| Quantity of UnPlanned Unavailability in Last 12 Months | 0.27 | 0.67 |
| Time Since Last UnPlanned Unavailability | 0.25 | 0.58 |

4 Modeling Strategy

4.1 Data Sampling

As the modeling strategy involves the creation of behavioral variables, there is statistical dependence among the examples, differently from typical classification problems. Therefore data division for modeling and testing the system should to be temporally disjoint in two blocks, as done in time series forecasting tasks [9] for more realistic performance assessment. The diagram in Fig. 2 below shows this division in time.

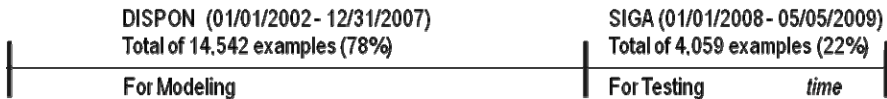


Fig. 2. Data partition along time for modeling and performance assessment of the system

This data partition took into account the change in the computational environment to represent the worst case in terms of performance assessment. In the modeling set, the target class (unavailability) represents 18.1% of the examples whereas, in the testing set, it represents only 10.5% of the examples. An additional difficulty is related to the differences in the way data were recorded before and after SIGA, which not even CHESF’s personnel can precisely assess.

The modeling data refer to the whole period before the SIGA system was deployed while the testing data have their target defined after SIGA’s deployment. The behavioral variables of the testing data, however, also capture historical information from the preceding period.

4.2 Logistic Regression and Model Ensemble

The modeling technique chosen was logistic regression for several interesting features it possesses being the quality and understandability of the solution produced and the small amount of data required the most relevant features for this work. Logistic regression has been successfully applied to binary classification problems, particularly to credit risk assessment [3], it does not require a validation set for over-fitting prevention and it presents explicitly the knowledge extracted from data in terms of statistically validated coefficients [10].

As preliminary experiments with different data samples showed a high variance in performance, it was clear that an ensemble of systems was necessary [11]. In this work, the ensemble consisting of 31 Logistic Regression models has reduced the system's variance and their median was taken as the response for each test example. This median approach had been adopted by the authors' teams since 2007 in PAKDD Data Mining Competition [12] and in NN3 Time Series Forecasting Competition [13]. As already stated, the modeling technique chosen was Logistic Regression due to its explicit coefficients and for not having the need of a validation set. For training the 31 models, 50% of the examples in the modeling data set were randomly sampled without replacement. These parameters were chosen by linear experimental project [14] with the ensemble size taking the values 31, 51 and 101 and the percentage taking the values 70% 60% and 50%.

5 Experimental Metrics, Results and Interpretation

5.1 Performance Metrics

As there was no criterion available yet for defining the decision threshold along the continuous output of the logistic regression ensemble, the performance assessment was carried out using two metrics for the whole decision domain (the score range): the maximum Kolmogorov-Smirnov distance (Max_KS) [7] and the Area Under the ROC Curve (AUC_ROC) [8]. The AUC_ROC metric is widely accepted for performance assessment of binary classification based on continuous output. Similar wide acceptance holds for the Max_KS within the business application domain.

Differently from its original purpose as a statistical non parametric tool for measuring the adherence of cumulative distribution functions (CDF) [7], in binary decision systems, the KS maximum distance is applied for assessing the lack of adherence between the data sets from the 2 classes, having the score as independent variable. The Kolmogorov-Smirnov Curves are the difference between the CDFs of the data sets of the two classes. The higher the curve, the better the system and the point of maximum value is particularly important in performance evaluation.

Another widely used tool is the Receiver Operating Characteristic Curve (ROC Curve) [8] whose plot represents the compromise between the true positive and the false positive example classifications based on a continuous output along all its possible decision threshold values (the score). The closer the ROC curve is to the upper left corner (optimum point), the better the decision system is. The focus is on assessing the performance throughout the whole X-axis range by calculating the area under the ROC curve (AUC) [8]. The bigger the area, the closer the system is to the optimum decision which happens with the AUC_ROC equal to one.

5.2 Results and Interpretation

Performance was assessed on the testing set which consisted of the out-of-sample data with 4,059 examples reserved for this purpose only. Fig. 3 shows the Kolmogorov-Smirnov curve with its Max_KS=0.341. Fig. 4 shows the ROC curve with its AUC_ROC=0.699.

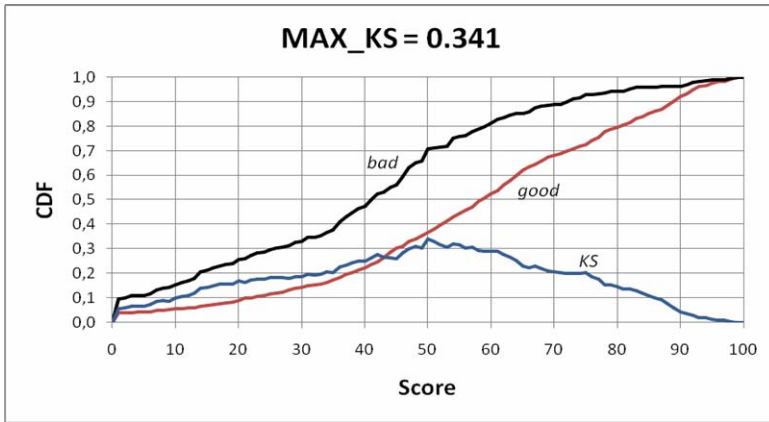


Fig. 3. Performance assessment by the Kolmogorov-Smirnov metric with Max_KS=0.341

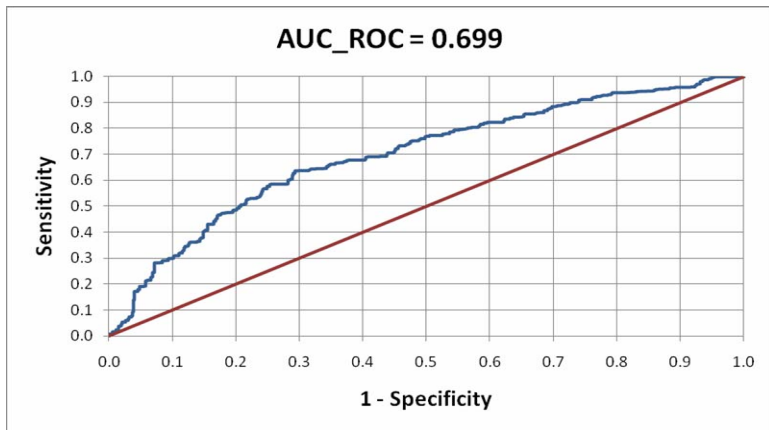


Fig. 4. Performance assessment by Area Under the ROC Curve metric with AUC_ROC=0.699

The curves are quite noisy probably because of the small amount of data in the testing set. There are only around 400 examples from the target class in this data set whose CDF is a very noisy curve (top plot in Fig. 3) whereas the non-target class (“good”) is a smooth curve. Even being noisy, the performance curves are consistent and present an improvement which will be useful, for CHESF, particularly considering that the testing set represents a worst case approximation.

6 Concluding Remarks

This paper has presented a domain driven data mining approach to the problem of Operational Function unavailability in the electrical power grid of one of the biggest power companies in Brazil - CHESF.

Different from statistical approaches, this innovative work has modeled the unavailability as a data mining binary decision problem with behavioral input variables. These variables were created by sliding windows of different sizes timed by the planned maintenance events which were labeled as “*bad*” when an unplanned unavailability occurred before its next planned maintenance.

An important advantage of this approach compared to the statistical ones is that it does not impose any constraint on the data distributions to be modeled. The only approximation made was to consider the planned unavailability’s cost negligible compared to that of an unplanned one; around 5% of the value.

It should be emphasized here that there is a big difference between the concepts of *approach* and *technique* which becomes clear when the statistical *technique* logistic regression is used within a domain driven data mining *approach* for modeling the whole problem as a sequence of rare events consolidated in RFI variables which capture sequential information in terms of Recency, Frequency and Impact.

The median of an ensemble of bagged logistic regression models has provided the unavailability’s risk estimating score and its coefficients made explicit the most relevant variables for each suggested decision.

Results of the experiments carried out on an out-of-sample test set have shown that the approach is viable for risk estimation. It attained a Max_KS=0.341 and AUC_ROC=0.699, in a worst case scenario.

After this approach’s validation, the testing data set has been included in the modeling data set and the system has been re-trained with the same procedure. Now, the system has just been put in operation and its performance will be monitored for the next six months when CHESF will be making pro-active maintenance based on the system predictions. Both the quality the solution and the availability of the power grid can lead to redesigning maintenance periods.

Several refinements still have to be made, particularly, those referring to the revenue losses caused by the penalties for power grid unavailability. This refinement can be made by considering the “losses” either in the modeling process or in the post-processing stage along with the risk estimating score produced by the decision support system. Also, the variable selection process should include multivariate techniques such as the variance inflation factor (VIF) [15].

References

1. ANEEL. Normative Resolution no. 270 (June 2007), <http://www.aneel.gov.br/cedoc/ren2007270.pdf>
2. CHESF. Companhia Hidro Elétrica do São Francisco, http://www.chesf.gov.br/acompanhia_visaoomissao.shtml
3. West, D.: Neural network credit scoring models. *Computers and Operations Research* 27, 1131–1152 (2000)
4. Jiang, T., Tuzhilin, A.: Improving Personalization Solutions through Optimal Segmentation of Customer Bases. *IEEE Trans. Knowledge and Data Eng.* 3(21), 1–16 (2009)
5. Cao, L.: Introduction to Domain Driven Data Mining. In: Cao, L., et al. (eds.) *Data Mining for Business Applications*, pp. 3–10 (2008)
6. Han, J., Kamber, M.: *Data Mining: Concepts and techniques*. Morgan Kaufmann, San Francisco (2006)

7. Conover, W.J.: *Practical Nonparametric Statistics*, 3rd edn. John Wiley & Sons, NY (1999)
8. Provost, F., Fawcett, T.: Robust Classification for Imprecise Environments. *J. Machine Learning* 3(42), 203–231 (2001)
9. Adya, M., Collopy, F.: How Effective are Neural Networks at Forecasting and Prediction? A Review and Evaluation. *J. of Forecasting* 17, 48–495 (1998)
10. Hilbe, J.M.: *Logistic Regression Models*. Chapman & Hall / CRC Press (2009)
11. Breiman, L.: Bagging predictors. *Machine Learning* 2(24), 123–140 (1996)
12. Adeodato, P.J.L., Vasconcelos, G.C., Arnaud, A.L., Cunha, R.C.L.V., Monteiro, D.S.M., Oliveira Neto, R.: The Power of Sampling and Stacking for the PAKDD-2007 Cross-Selling Problem. *Int. J. of Data Warehousing and Mining (IJDWM)* 4, 22–31 (2008)
13. Adeodato, P.J.L., Vasconcelos, G.C., Arnaud, A.L., Cunha, R.C.L.V., Monteiro, D.S.M.P.: MLP ensembles improve long term prediction accuracy over single networks. *Int. J. of Forecasting* (2010) (to appear)
14. Jain, R.: *The Art of Computer Systems Performance Analysis Techniques for Experimental Design Measurements Simulation and Modeling*. John Wiley & Sons, New York (1991)
15. Kutner, M., Nachtsheim, C., Neter, J.: *Applied Linear Regression Models*, 4th edn. McGraw-Hill / Irwin (2004)

Social Order in Hippocratic Multi-Agent Systems

Ludivine Crépin¹, Yves Demazeau², Olivier Boissier³, and François Jacquenet⁴

¹ Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier,
161 rue Ada 34392 Montpellier Cedex France

`Ludivine.Crepin@lirmm.fr`

² Laboratoire d'Informatique de Grenoble - CNRS, 110 avenue de la chimie 38000
Grenoble cedex France

`Yves.Demazeau@imag.fr`

³ Ecole Nationale Supérieure des Mines de Saint-Etienne - Centre G2I,
158 cours Fauriel 42000 Saint-Etienne cedex France

`Olivier.Boisser@emse.fr`

⁴ Université Jean Monnet - Laboratoire Hubert Curien - CNRS, 18 rue Benoit
Lauras 42000 Saint Etienne Cedex France

`Francois.Jacquenet@univ-st-etienne.fr`

Abstract. In multi-agent applications, users delegate their sensitive data to autonomous agents that interact with other autonomous agents. In this context, privacy preservation is an important topic. In previous work, considering this problem, we have proposed the Hippocratic Multi-Agent System model (HiMAS). In this paper, we focus on the regulation of agents behavior with respect to privacy management in this model. We present a social order approach based on trust and reputation that install a decentralised regulation of privacy management in HiMAS systems.

Keywords: Hippocratic Multi-Agent System, Privacy Preservation, Social Order, Trust, Reputation.

1 Introduction

In user centered multi-agent systems, agents assist users by managing data that they delegate. To tackle the privacy preservation problems that are induced by automatic processing of sensitive data, we proposed the model of Hippocratic Multi-Agent System (HiMAS) [6]. HiMAS proposes a MAS framework integrating the management of *private sphere* with nine principles to preserve privacy.

This model requires to specify and implement regulation mechanisms for the detection of suspicious agents (i.e. agents that violate the private sphere). The private sphere and all relative information are **personal** and **subjective**. By this way, external regulation mechanisms like PONDER [10] or the electronic institutions like ISLANDER [8] for example are not appropriate because this kind of systems requires that an entity knows every communicated sensitive data, but this is in fact a privacy violation.

In this article, we integrate in HiMAS an internal regulation mechanism that protects the private sphere: a social order based on trust and reputation. We propose a generic framework for trust model in order to transform a social order from hippocratic point of view for privacy preservation.

Section 2 presents the HiMAS model. Section 3 focuses on the trust model installing the social order in the system. In section 4 we present a hippocratic social order that regulates agents behavior with regards to privacy. To do so we integrate the HiMAS principles in the process and check their respect for all exchanged sensitive data. The experimental validation of our proposition is presented in the next section. We finally conclude and provide some perspectives.

2 Hippocratic Multi-Agent Systems (HiMAS)

In this section we briefly recall the basic principles of the model of Hippocratic Multi-Agent Systems [6] that we have have proposed for privacy preservation.

2.1 Private Sphere, Consumer and Provider

In a HiMAS, the *private sphere* is modeled by the set of sensitive data to be preserved with their management rules. A HiMAS agent can play two roles in the context of data communication (called *sensitive data transaction*): it may be a *provider* that sends sensitive data to another agent that is called the *consumer*.

2.2 Nine Normative Principles to Preserve Privacy

In order to preserve privacy, a HiMAS imposes agents to respect nine normative principles inspired by the hippocratic databases [1]:

- **1. Purpose specification:** The provider must know the objectives of the sensitive data transaction.
- **2. Consent:** Each sensitive data transaction requires the provider's consent.
- **3. Limited collection:** The consumer commits to cutting down the amount of data for realizing its objectives to a minimum.
- **4. Limited use:** The consumer commits to using sensitive provider's data only to satisfy the objectives that it has specified and nothing more.
- **5. Limited disclosure:** The consumer commits to only disclosing the sensitive data needed to reach its objectives.
- **6. Limited retention:** The consumer commits to retaining sensitive data only for the minimum amount of time needed to perform its objectives.
- **7. Safety:** The system must guarantee sensitive data safety during storage and transactions.
- **8. Openness:** The transmitted sensitive data must remain accessible to the provider during the retention time.
- **9. Compliance:** Each agent should be able to check the obedience to the previous principles.

Seven of these principles (1.-2.-3.-4.-5.-6.-8.) are embedded in a sensitive data transaction protocol [5] that we briefly present in the next subsection. In this article, we focus on the *compliance principle*, that we propose to model with the social order based on trust and reputation (section 3).

2.3 Sensitive Data Transaction Protocol

The sensitive data transaction protocol [5] allows HiMAS agents to check the intention of the consumer in relation to the required data thanks to a content language. This content language defines all the possibilities for the sensitive data manipulations in terms of collection, use, disclosure, retention and openness regarding the respect of the private sphere after a transaction. The HiMAS agents use this content language to build their *policy* (resp. *preference*) when they endorse the consumer (resp. provider). Policy and preference specify the objectives of the transaction, the disclosure list, the future uses and the retention time of the sensitive data.

This protocol allows the HiMAS agents to detect suspicious behaviors: if an agent does not respect the manipulations specified by the content language, it is considered as suspicious and the transaction is cancelled. The provider sends required sensitive data to the consumer when there is an agreement between their policy and preference, otherwise the transaction is also cancelled.

After a sensitive data transaction, the agents attach the policy to the communicated sensitive data and the compliance principle can act: each agent should be able to check the respect of the policy.

3 Trust and Reputation Model for Social Order

The social order proposed in [3] represents the collaboration between agents based on trust and more particularly on a recommendation process, using propagated reputation [2], an internal behavior regulation mechanism. The respect of the private sphere does not allow external regulation mechanism due to its personal and subjective aspect [6]. In this direction, we need to choose a trust model that takes these two parameters into account and that also allows to build a hippocratic social order with regards to privacy preservation.

3.1 Multi-agent Trust Model

Our research context being the user centered multi-agent systems, the user is usually delegating her private sphere to an autonomous agent but also to all the regulation process. In order to adapt our model for every applications, we propose to extend the Castelfranchi and Falcone model [4] that is designed for a general context of multi-agent systems, to introduce a hippocratic social order into a

¹ This fact imposes only a local vision of the policy and the preference for the agents, there is no possibility to get a global vision for the agents society without private sphere violation.

HiMAS because it does not depend on the domain and this makes its adaptation easier to any context. Moreover, it is a cognitive model that corresponds to a user centered approach [9].

3.2 Trust Model for Hippocratic Social Order

Castelfranchi and Falcone defines in [4] the concept of trust as a task delegation between agents: an agent delegates a task to another one only if a trust relationship exists between these two agents.

Trust is then studied as a set of mental states that are based on three kinds of information in order to establish a trust relationship according to a given context (or a given action) Ω : the direct experiences, the recommendations and the systemic trust (i.e. direct, propagated and stereotyped reputation). In a HiMAS, Ω represents the objectives of the sensitive data transaction.

In order to establish a trust relationship with an agent j , an agent i compares its trust belief of a trust function to a threshold where 0 means no trust and 1 an absolute trust for j in the given context Ω .

Applying the compliance principle requires to give HiMAS agents the ability to check the respect of the other principles. To do so, we make a first strong hypothesis: each HiMAS agent respects the compliance principle by denouncing every suspicious behavior it detects according to the sensitive data policy.

The trust model is grounded to a punishment model: only the privacy violations change the trust belief negatively. A trust relationship can be destroyed but cannot be rebuilt because no reward system is applied, so that the trust judgement can only decrease.

The social order we propose allows the HiMAS agents to pass a judgement on the reliability of the other agents according the four kinds of constraints, that represent the facet of trust belief, specified in a policy: use, disclosure, retention and openness. The three information sources must be linked to this four facets, noted f , the consumer and the sensitive data transaction context. We define the compilation of these information sources as:

- $DoDR_{c,\Omega,f}$ (*Degree of Direct Reputation*): the provider belief about all the direct experiences (policies respect) with the consumer c according to the facet f and the context Ω ;
- $DoPR_{c,\Omega,f}$ (*Degree of Propagated Reputation*): the provider belief in relation to the recommendation about the respect of the policies for the consumer c according to the facet f and the context Ω ;
- $DoSR_{c,\Omega,f}$ (*Degree of Stereotyped Reputation*): the provider belief in relation to the characteristics of the consumer f on its capacity to respect its policy according to the facet f and the context Ω .

In order to get a trust value [4], the HiMAS agents should determine two beliefs that belong to the interval $[0,1]$:

- $DoA_{c,\Omega,f}$ (*Degree of Ability*) = $F^A(DoDR_{c,\Omega,f}, DoPR_{c,\Omega,f})$ where F^A is the compilation of $DoDR$ and $DoPR$.

- $DoW_{c,\Omega,f}$ (*Degree of Willingness*) = $F^W(DoPR_{c,\Omega,f}, DoST_{c,\Omega,f})$ where F^W is the compilation of $DoPR$ and $DoST$.

The trust belief a provider assigns to a consumer c according to the facet f and the given context Ω is determined by the following function:

$$DoT_{c,\Omega,f}(\textit{Degree of Trust}) = F(F_{c,\Omega,f}^A, F_{c,\Omega,f}^W)$$

The last step to establish a trust relationship consists in combining of the trust belief for each facet:

$$DoT_{c,\Omega} = f'(f_{c,\Omega,f_1}^A \cdot \cdot f_{c,\Omega,f_n}^A, f_{c,\Omega,f_1}^W \cdot \cdot f_{c,\Omega,f_n}^W)$$

4 Hippocratic Social Order

The extension of the model proposed in [4] we propose allows us to introduce a hippocratic social order in HiMAS in order to model the compliance principle.

This task requires first to establish a trust relationship with regards to the HiMAS model. Then we need to model the basis of the hippocratic social order after a sensitive data transaction between a consumer and a provider. These two steps allow us to implement our proposition and to determine the possible impacts on the HiMAS agents. We do not evaluate the trust model performance but its use in order to model the compliance principle.

4.1 Hippocratic Trust Relationship

Establishing a trust relationship allows to preserve the private sphere by detecting suspicious agents. This relationship allows agents of a HiMAS to pass a judgement on consumers reliability about the respect of their policies. We consider that all the trust information, and more particularly the propagated reputations, are sensitive data and must be protected by the HiMAS principles, because these data are in direct relation to the users represented by the agent.

To guarantee the respect of the private sphere, we propose to integrate the process of trust building and management in a hippocratic model that will be independent of the trust model that is used and is generic for each model that manage the propagated reputations. This hippocratic model allows to preserve trust data thanks to the nine HiMAS normative principles.

In order to respect the protocol we have proposed in [5], we extend the content language with the specific case of the propagated reputations with a specific objective "social order". This objective defines that the received data are the propagated reputations and that these kinds of data must only be used in order to pass a judgement on another agent or to revise an old judgement. Moreover, propagated reputations must not be disclosed and must be accessible only during the step of the corresponding process.

With the hippocratic social order, HiMAS agents should exchange propagated reputations. With this protocol, we impose to agency to provide their own trust data and to not forward those of another agent.

4.2 Sensitive Data Transaction and Compliance Principle

HiMAS agents being now able to pass a judgment on the consumers reliability, we need to introduce trust in the sensitive data transaction protocol [5].

Taking into account this aspect introduces a new step in the reasoning mechanism of the provider (see Figure 1 in red). The provider uses its trust beliefs to decide about the acceptance of the consumer policy. In case this level is correct, the provider decide to continue the transaction, else to cancel the transaction.

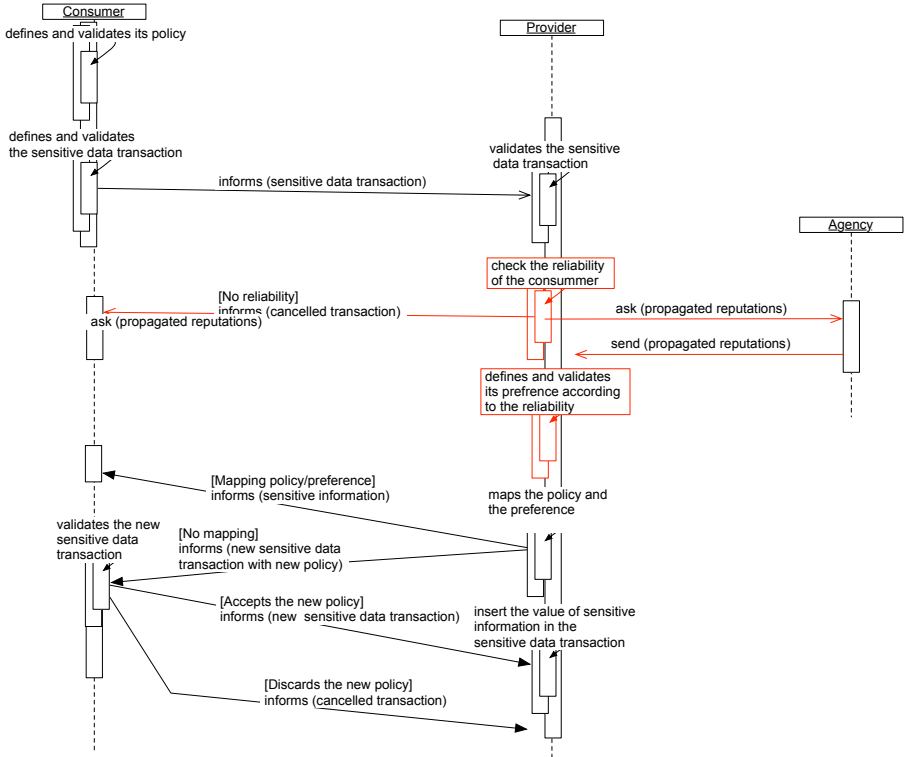


Fig. 1. Sensitive data transaction integrating hippocratic social order

5 Experiments

5.1 Scenario and Parameters

To evaluate our proposition, we chose a scenario related to a specific domain: the management of calendars [7]. Each user is represented by an agent that manages her calendar. All the agents share and disclose randomly the meetings using the sensitive data transaction protocol proposed in [5].

After each sensitive data transaction, the consumer links the corresponding policy to the sensitive data it receives and includes it in its private sphere. When the consumer becomes a provider and when it sends this sensitive data, it also respects this policy and sends it as a preference that the new consumer must respect. By this way, the new consumer can check the reliability of the past consumer. If the policy has been violated, the new consumer sends a warning to the agency and the trust level of the suspicious decreases.

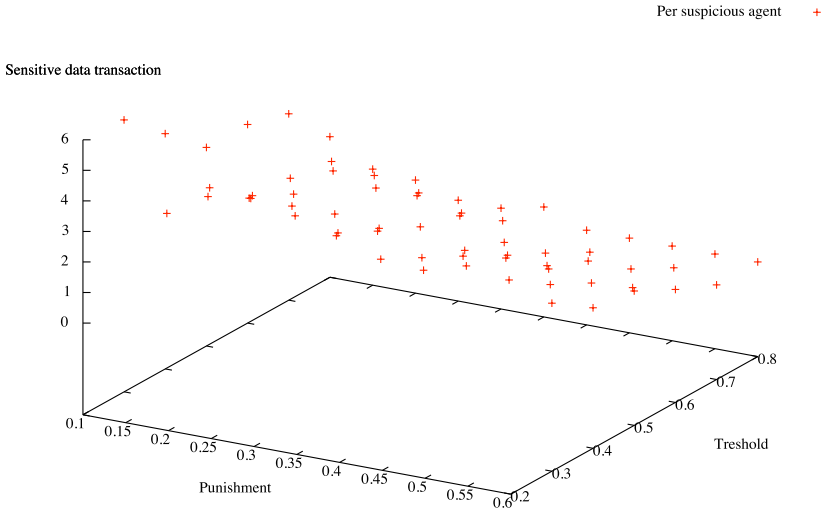


Fig. 2. Average of required sensitive data transactions per agent with the suspicious agent for the detection of the suspicious agent according to the punishment and threshold value

During the initialization of an agent, its calendar is randomly created with a set of meetings. The $DoST_{c,\Omega,f}$, $DoDR_{c,\Omega,f}$ and $DoPR_{c,\Omega,f}$ for each agent and for each facet get the value of 1. The policies and preferences are chosen randomly in the ontology of the content language presented in [5]. A suspicious agent is an agent that violates its policy every time for every facet. This agent is banished of the agency when a minority of blocking (a third of the agency) considers it as suspicious. In order to not banish an agent too much quickly² (see Figure 2), we have fixed the decrease of reputation to 0,15. The threshold deciding when to destroy a trust relationship is fixed to 0,5.

In order to test our hippocratic social order, we experiment 20 times this scenario with different sets of agents with one suspicious agent and according to three kinds of networks: a social network (no dependance constraint between

² An agent that violates a policy only one times is not always suspicious. Indeed it should make this violation in order to realize one specific objective.

agents), a tree network (an agent can interact with its relative and its son) and a layered network (an agent can interact with its relatives and its sons).

The second kind of experiments focuses on the number of suspicious agents. In a set of 50 agents organized in social networks, we introduce 1, 5, 12, 18 and 25 suspicious agents in the agency. By this way, we study the limitations of our proposition according to the percentage of suspicious agents.

5.2 Results

Network topology. In social networks, the detection of a suspicious agent is not related to the number of agents in the agency. Indeed, only about 2,5 sensitive data transactions per agent are required with the suspicious agent for its detection (see Figure 3).

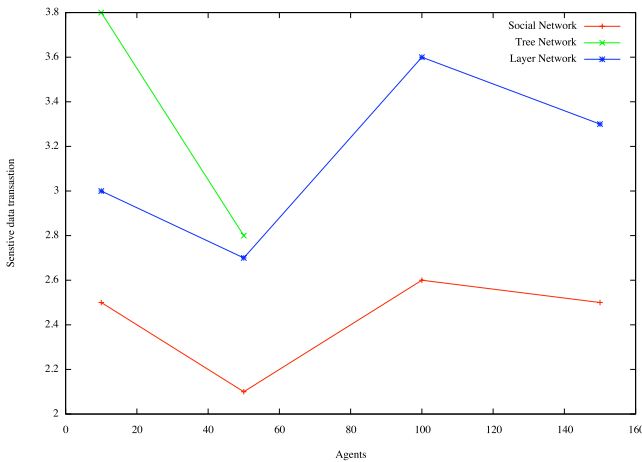


Fig. 3. Average of required sensitive data transactions per agent with the suspicious agent for the detection of the suspicious agent according to the networks topology

For tree networks, due to an important requirement of memory because of the number of communications, our experiments could not exceed a society of 50 agents. In fact, a simulation with 50 agents requires around 1940 transactions that is approach approximately the required number for 150 agents in the social network. However, even if the number of sensitive data transactions increases, the required number of transactions in order to detect suspicious agents remains at the same level that for the social networks (Figure 3).

In the layered networks, the number of sensitive data transactions is higher than in the social networks (see Figure 3), but the required number of sensitive data transactions with the suspicious agents is about 3,5 transactions per agent (see Figure 3) and so the performance is the same than in the social networks.

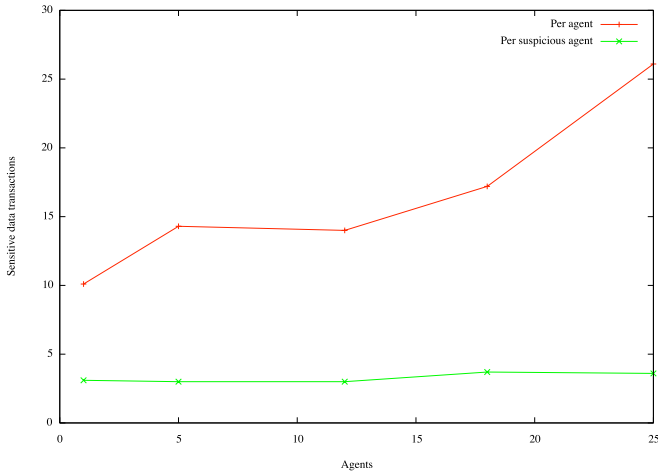


Fig. 4. Average of required sensitive data transactions per agent for the detection of the suspicious agent according to the number of suspicious agents

To sum up, we can conjecture that the number of agents and the network topology do not really influence the detection of the suspicious agent: each agent needs about 3 sensitive data transactions for its detection.

Number of suspicious agents. We can notice that the increase of the number of suspicious agents in a agency can affect the suspicious behavior detection. Indeed, as shown at Figure 4 the results are in the same interval as the ones previously presented until a percentage of 30% of the agency: the average of sensitive transaction for the suspicious agent detection per agent and per suspicious agent is the same for only one agent. After this percentage, the hippocratic social order becomes less powerful because the average of the sensitive data transaction per agent become more important.

6 Conclusions and Perspectives

In order to model and implement the compliance principle defined in the HiMAS model, we have used a trust model that allows us to enforce privacy preservation without any global view of policies for the agency. The trust model that we propose is an extension of Castelfranchi and Falcone model [4]. It includes the preferences of the users for the trust management using reputations.

We have also investigated privacy preservation in the trust process by considering trust information as sensitive data. By this way, we extend the content language proposed in [5] in order to take the objective of social order into account and introduce trust in the sensitive data protocol to check consumer reliability.

Our perspectives first focus on the implementation of our proposition in other trust models in order to determine their influence on privacy preservation.

In another direction of work, we propose to extend a multi-agent system application dealing with decentralized calendar management [7] that already uses the sensitive data transaction protocol [5] by implementing the hippocratic social order.

Acknowledgments. This work is supported by the Web Intelligence project, funded by the ISLE cluster of the Rhône-Alpes region.

References

1. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Hippocratic databases. In: Proceedings of the International Conference on Very Large Data Bases, pp. 143–154. Morgan Kaufmann, San Francisco (2002)
2. Casare, S.J., Sichman, J.S.: Towards a functional ontology of reputation. In: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 505–511. ACM, New York (2005)
3. Castelfranchi, C.: Engineering social order. In: Omicini, A., Tolksdorf, R., Zambonelli, F. (eds.) ESAW 2000. LNCS (LNAI), vol. 1972, pp. 1–18. Springer, Heidelberg (2000)
4. Castelfranchi, C., Falcone, R.: Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In: Proceedings of the International Conference on Multiagent Systems, pp. 72–79. IEEE Computer Society, Los Alamitos (1998)
5. Crépin, L., Demazeau, Y., Boissier, O., Jacquenet, F.: Sensitive data transaction in hippocratic multi-agent systems. In: Artikis, A., Picard, G., Vercouter, L. (eds.) ESAW 2008. LNCS, vol. 5485, pp. 85–101. Springer, Heidelberg (2009)
6. Crépin, L., Vercouter, L., Jacquenet, F., Demazeau, Y., Boissier, O.: Hippocratic multi-agent systems. In: Proceedings of the 10th International Conference of Enterprise Information Systems, pp. 301–308 (2008)
7. Demazeau, Y., Melaye, D., Verrons, M.-H.: A decentralized calendar system featuring sharing, trusting and negotiating. In: Ali, M., Dapoigny, R. (eds.) IEA/AIE 2006. LNCS (LNAI), vol. 4031, pp. 731–740. Springer, Heidelberg (2006)
8. Esteva, M., de la Cruz, D., Sierra, C.: Islander: an electronic institutions editor. In: Proceedings of the International Joint Conference on Autonomous Agents & Multiagent Systems, pp. 1045–1052. ACM, New York (2002)
9. Lacomme, L., Demazeau, Y., Camps, V.: Personalization of a trust network. In: Proceedings of the International Conference on Agents and Artificial Intelligence, pp. 408–415. IEEE/ACM (2009)
10. Lupu, E., Sloman, M., Dulay, N., Damianou, N.: Ponder: Realising enterprise viewpoint concepts. In: Proceedings of the International Enterprise Distributed Object Computing Conference, pp. 66–75. IEEE Computer Society, Los Alamitos (2000)

Building an Electronic Market System

Elaine Lawrence and John Debenham

QCIS, FEIT, University of Technology, Sydney, Australia
elaine@it.uts.edu.au

Abstract. An electronic market system is predicated on three technologies: data mining, intelligent trading agents and virtual institutions in which informed trading agents can trade securely both with each other and with human agents in a natural way. This paper describes a demonstrable prototype electronic market that integrates these three technologies and is available on the World Wide Web. This is part of a larger project that aims to make informed automated trading a reality.

1 Introduction

Electronic market trading involves the complete process of: need identification, product brokering, supplier brokering, offer-exchange, contract negotiation, and contract execution [1], as well as the ability to model business relationships. Three core technologies are needed to support electronic markets:

- trading agents — intelligent agents that are designed to operate in tandem with the real-time information flows received from data mining systems.
- data mining — real-time data mining technology to tap information flows from the marketplace and the World Wide Web, and to deliver timely information at the right granularity.
- virtual institutions — virtual places on the World Wide Web in which informed trading agents can trade securely both with each other and with human agents in a natural way — not to be confused with the term “virtual organisations” as used in Grid computing.

This paper describes an e-trading system that integrates these three technologies. The e-Market Framework is available on the World Wide Web [1]. This project aims to make informed automated trading a reality, and develops further the “Curious Negotiator” framework [2]. This work does not address all of the issues in automated trading. For example, the work relies on developments in: XML and semantic web, secure data exchange, value chain management and financial services.

2 Data Mining

We have designed information discovery and delivery agents that utilise text and network data mining for supporting real-time negotiation. This work has

¹ <http://e-markets.org.au>

addressed the central issues of extracting relevant information from different on-line repositories with different formats, with possible duplicative and erroneous data. That is, we have addressed the central issues in extracting information from the World Wide Web. Our mining agents understand the influence that extracted information has on the subject of negotiation and takes that in account.

Real-time embedded data mining is an essential component of the proposed framework. In this framework the trading agents make their informed decisions, based on utilising two types of information: First, information extracted from the negotiation process (i.e. from the exchange of offers). Second, information from external sources, extracted and provided in condensed form.

The embedded data mining system provides the information extracted from the external sources. The system complements and services the information-based architecture developed in [3] and [4]. The data mining system initially constructs data sets that are “focused” on requested information. From the vast amount of information available in electronic form, we need to filter the information that is relevant to the information request. In our example, this will be the news, opinions, comments, white papers related to five models of digital cameras. Technically, the automatic retrieval of the information pieces utilises the universal news bot architecture presented in [5]. Developed originally for news sites only, the approach is currently being extended to discussion boards and company white papers.

The “focused” data set is dynamically constructed in an iterative process. The data mining agent constructs the news data set according to the concepts in the query. Each concept is represented as a cluster of key terms (a term can include one or more words), defined by the proximity position of the frequent key terms. On each iteration the most frequent (terms) from the retrieved data set are extracted and considered to be related to the same concept. The extracted keywords are resubmitted to the search engine. The process of query submission, data retrieval and keyword extraction is repeated until the search results start to derail from the given topic.

The set of topics in the original request is used as a set of class labels. In our example we are interested in the evidence in support of each particular model camera model. A simple solution is for each model to introduce two labels — positive opinion and negative opinion, ending with ten labels. In the constructed “focused” data set, each news article is labelled with one of the values from this set of labels. An automated approach reported in [5] extends the tree-based approach proposed in [6].

Once the set is constructed, building the “advising model” is reduced to a classification data mining problem. As the model is communicated back to the information-based agent architecture, the classifier output should include all the possible class labels with an attached probability estimates for each class. Hence, we use probabilistic classifiers (e.g. Naïve Bayes, Bayesian Network classifiers [7] without the min-max selection of the class output [e.g., in a classifier based on Naïve Bayes algorithm, we calculate the posterior probability $\mathbb{P}_p(i)$ of each class $c(i)$ with respect to combinations of key terms and then return the tuples

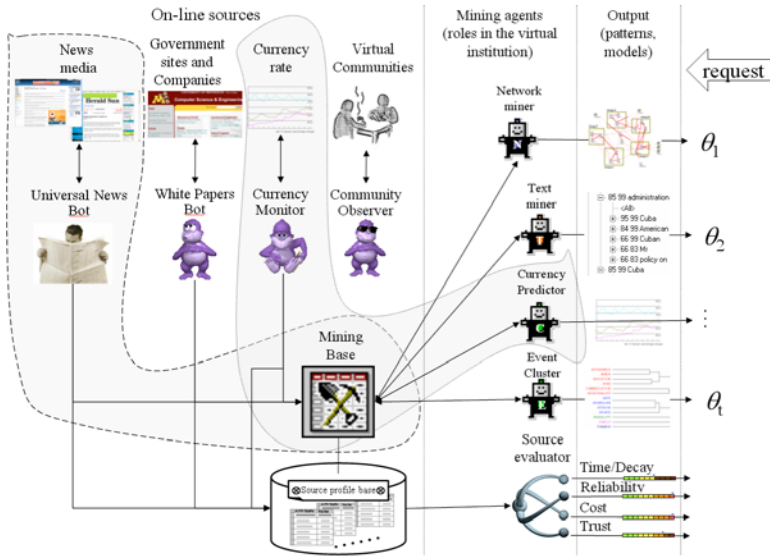


Fig. 1. The architecture of the agent-based data mining system

$\langle c(i), \mathbb{P}_p(i) \rangle$ for all classes, not just the one with maximum $\mathbb{P}_p(i)$. In the case when we deal with range variables the data mining system returns the range within which is the estimated value. For example, the response to a request for an estimate of the rate of change between two currencies over specified period of time will be done in three steps: (i) the relative focused news data set will be updated for the specified period; (ii) the model that takes these news in account is updated, and; (iii) the output of the model is compared with requested ranges and the matching one is returned. The details of this part of the data mining system are presented in [8]. The currently used model is a modified linear model with an additional term that incorporates a news index I_{news} , which reflects the news effect on exchange rate. The current architecture of the data mining system in the e-market environment is shown in Figure 1. The $\{\theta_1, \dots, \theta_t\}$ denote the output of the system to the information-based agent architecture. In addition, the data mining system provides parameters that define the “quality of the information”, including:

- the time span of the “focused” data set, defined by the eldest and the latest information unit);
- estimates of the characteristics of the information sources, including reliability, trust and cost, that then are used by the information-based agent architecture.

3 Trading Agents

We have designed a new agent architecture founded on information theory. These “information-based” agents operate in real-time in response to market

information flows. We have addressed the central issues of trust in the execution of contracts, and the reliability of information [4]. Our agents understand the value of building business relationships as a foundation for reliable trade. An inherent difficulty in automated trading — including e-procurement — is that it is generally multi-issue. Most of the work on multi-issue negotiation has focussed on one-to-one bargaining — for example [9]. There has been rather less interest in one-to-many, multi-issue auctions — [10] analyzes some possibilities. The main focus of our agents is their information and their strength of belief in its integrity [11]. If their information is sufficiently certain then they may be able to estimate a utility function and to operate rationally in the accepted sense. However an agent may also be prepared to develop a semi-cooperative, non-utilitarian relationship with a trusted partner.

An agent called Π is the subject of this discussion. Π engages in multi-issue negotiation with a set of other agents: $\{\Omega_1, \dots, \Omega_o\}$. The foundation for Π 's operation is the information that is generated both by and because of its negotiation exchanges. Any message from one agent to another reveals information about the sender. Π also acquires information from the environment — including general information sources — to support its actions. Π uses ideas from information theory to process and summarize its information. Π 's aim may not be “utility optimization” — it may not be aware of a utility function. If Π *does* know its utility function *and* if it aims to optimize its utility *then* Π may apply the principles of game theory to achieve its aim. The information-based approach does not reject utility optimization — in general, the selection of a goal and strategy is secondary to the processing and summarizing of the information.

In addition to the information derived from its opponents, Π has access to a set of information sources $\{\Theta_1, \dots, \Theta_t\}$ that may include the marketplace in which trading takes place, and general information sources such as news-feeds accessed via the Internet. Together, Π , $\{\Omega_1, \dots, \Omega_o\}$ and $\{\Theta_1, \dots, \Theta_t\}$ make up a multiagent system. The integrity of Π 's information, including information extracted from the Internet, will decay in time. The way in which this decay occurs will depend on the type of information, and on the source from which it was drawn. Little appears to be known about how the integrity of real information, such as news-feeds, decays, although its validity can often be checked — “Is company X taking over company Y?” — by proactive action given a cooperative information source Θ_j . So Π has to consider how and when to refresh its decaying information.

Π triggers a goal, $g \in \mathcal{G}$, in two ways: first in response to a message received from an opponent $\{\Omega_i\}$ “I offer you €1 in exchange for an apple”, and second in response to some need, $\nu \in \mathcal{N}$, “goodness, we’ve run out of coffee”. In either case, Π is motivated by a need — either a need to strike a deal with a particular feature (such as acquiring coffee) or a general need to trade. Π 's goals could be short-term such as obtaining some information “what is the time?”, medium-term such as striking a deal with one of its opponents, or, rather longer-term such as building a (business) relationship with one of its opponents. So Π has a trigger mechanism T where: $T : \{\mathcal{X} \cup \mathcal{N}\} \rightarrow \mathcal{G}$.

For each goal that Π commits to, it has a mechanism, G , for selecting a strategy to achieve it where $G : \mathcal{G} \times \mathcal{M} \rightarrow \mathcal{S}$ where \mathcal{S} is the strategy library. A strategy s maps an information base into an action, $s(\mathcal{Y}^t) = z \in \mathcal{Z}$. Given a goal, g , and the current state of the social model m^t , a strategy: $s = G(g, m^t)$. Each strategy, s , consists of a *plan*, b_s and a *world model* (construction and revision) *function*, J_s , that constructs, and maintains the currency of, the strategy's *world model* W_s^t that consists of a set of probability distributions. A *plan* derives the agent's next action, z , on the basis of the agent's world model for that strategy and the current state of the social model: $z = b_s(W_s^t, m^t)$, and $z = s(\mathcal{Y}^t)$. J_s employs two forms of entropy-based inference:

- Maximum entropy inference, J_s^+ , first constructs an *information base* \mathcal{I}_s^t as a set of sentences expressed in \mathcal{L} derived from \mathcal{Y}^t , and then from \mathcal{I}_s^t constructs the world model, W_s^t , as a set of complete probability distributions.
- Given a prior world model, W_s^u , where $u < t$, minimum relative entropy inference, J_s^- , first constructs the incremental information base $\mathcal{I}_s^{(u,t)}$ of sentences derived from those in \mathcal{Y}^t that were received between time u and time t , and then from W_s^u and $\mathcal{I}_s^{(u,t)}$ constructs a new world model, W_s^t .

The illocutions in the communication language \mathcal{C} include information, $[info]$. The information received from general information sources will be expressed in terms defined by Π 's ontology. The procedure for updating the world model as $[info]$ is received follows. If at time u , Π receives a message containing $[info]$ it is time-stamped and source-stamped $[info]_{(\Omega, \Pi, u)}$, and placed in a repository \mathcal{Y}^t . If Π has an active plan, s , with model building function, J_s , then J_s is applied to $[info]_{(\Omega, \Pi, u)}$ to derive constraints on some, or none, of Π 's distributions. The extent to which those constraints are permitted to effect the distributions is determined by a value for the *reliability* of Ω , $R^t(\Pi, \Omega, O([info]))$, where $O([info])$ is the ontological context of $[info]$.

In the absence of new $[info]$ the integrity of distributions decays. If $D = (q_i)_{i=1}^n$ then we use a geometric model of decay:

$$q_i^{t+1} = (1 - \rho^D) \times d_i^D + \rho^D \times q_i^t, \text{ for } i = 1, \dots, n \quad (1)$$

where $\rho^D \in (0, 1)$ is the decay rate. This raises the question of how to determine ρ^D . Just as an agent may know the decay limit distribution it may also know something about ρ^D . In the case of an information-overfed agent there is no harm in conservatively setting ρ^D "a bit on the low side" as the continually arriving $[info]$ will sustain the estimate for D .

We now describe how new $[info]$ is imported to the distributions. A single chunk of $[info]$ may effect a number of distributions. Suppose that a chunk of $[info]$ is received from Ω and that Π attaches the epistemic belief probability $R^t(\Pi, \Omega, O([info]))$ to it. Each distribution models a facet of the world. Given a distribution $D^t = (q_i^t)_{i=1}^n$, q_i^t is the probability that the possible world ω_i for D is the true world for D . The effect that a chunk $[info]$ has on distribution D is to enforce the set of linear constraints on D , $J_s^D([info])$. If the constraints $J_s^D([info])$ are taken by Π as valid then Π could update D to the posterior

distribution $(p_i^{[info]})_{i=1}^n$ that is the distribution with least relative entropy with respect to $(q_i^t)_{i=1}^n$ satisfying the constraint:

$$\sum_i \{p_i^{[info]} : J_s^D([info]) \text{ are all } \top \text{ in } \omega_i\} = 1. \tag{2}$$

But $R^t(\Pi, \Omega, O([info])) = r \in [0, 1]$ and Π should only treat the $J_s^D([info])$ as valid if $r = 1$. In general r determines the extent to which the effect of $[info]$ on D is closer to $(p_i^{[info]})_{i=1}^n$ or to the prior $(q_i^t)_{i=1}^n$ distribution by:

$$p_i^t = r \times p_i^{[info]} + (1 - r) \times q_i^t \tag{3}$$

But, we should only permit a new chunk of $[info]$ to influence D if doing so gives us new information. For example, if 5 minutes ago a trusted agent advises Π that the interest rate will go up by 1%, and 1 minute ago a very unreliable agent advises Π that the interest rate may go up by 0.5%, then the second unreliable chunk should not be permitted to ‘overwrite’ the first.

Information reliability. We estimate $R^t(\Pi, \Omega, O([info]))$ by measuring the error in information. Π ’s plans will have constructed a set of distributions. We measure the ‘error’ in information as the error in the effect that information has on each of Π ’s distributions. Suppose that a chunk of $[info]$ is received from agent Ω at time s and is verified at some later time t . For example, a chunk of information could be “the interest rate will rise by 0.5% next week”, and suppose that the interest rate actually rises by 0.25% — call that correct information $[fact]$. What does all this tell agent Π about agent Ω ’s reliability? Consider one of Π ’s distributions D that is $\{q_i^s\}$ at time s . Let $(p_i^{[info]})_{i=1}^n$ be the minimum relative entropy distribution given that $[info]$ has been received as calculated in Eqn. 2, and let $(p_i^{[fact]})_{i=1}^n$ be that distribution if $[fact]$ had been received instead. Suppose that the reliability estimate for distribution D was R_D^s . This section is concerned with what R_D^s should have been in the light of knowing *now*, at time t , that $[info]$ should have been $[fact]$, and how that knowledge effects our current reliability estimate for D , $R^t(\Pi, \Omega, O([info]))$.

The idea of Eqn. 3, is that the current value of r should be such that, *on average*, $(p_i^s)_{i=1}^n$ will be seen to be “close to” $(p_i^{[fact]})_{i=1}^n$ when we eventually discover $[fact]$ — no matter whether or not $[info]$ was used to update D . That is, given $[info]$, $[fact]$ and the prior $(q_i^s)_{i=1}^n$, calculate $(p_i^{[info]})_{i=1}^n$ and $(p_i^{[fact]})_{i=1}^n$ using Eqn. 2. Then the *observed reliability* for distribution D , $R_D^{([info]||[fact])}$, on the basis of the verification of $[info]$ with $[fact]$ is the value of r that minimises the Kullback-Leibler distance between $(p_i^s)_{i=1}^n$ and $(p_i^{[fact]})_{i=1}^n$:

$$\arg \min_r \sum_{i=1}^n (r \cdot p_i^{[info]} + (1 - r) \cdot q_i^s) \log \frac{r \cdot p_i^{[info]} + (1 - r) \cdot q_i^s}{p_i^{[fact]}}$$

If $E^{[info]}$ is the set of distributions that $[info]$ effects, then the overall *observed reliability* on the basis of the verification of $[info]$ with $[fact]$ is:

$$R^{([info]||[fact])} = 1 - \left(\max_{D \in E^{[info]}} |1 - R_D^{([info]||[fact])}| \right)$$

Then for each ontological context o_j , at time t when, perhaps, a chunk of $[info]$, with $O([info]) = o_k$, may have been verified with $[fact]$:

$$R^{t+1}(II, \Omega, o_j) = (1 - \rho) \times R^t(II, \Omega, o_j) + \rho \times R^{([info][fact])} \times \text{Sem}(o_j, o_k) \quad (4)$$

where $\text{Sem}(\cdot, \cdot) : O \times O \rightarrow [0, 1]$ measures the semantic distance between two sections of the ontology, and ρ is the learning rate. Over time, II notes the ontological context of the various chunks of $[info]$ received from Ω and over the various ontological contexts calculates the relative frequency, $P^t(o_j)$, of these contexts, $o_j = O([info])$. This leads to an overall expectation of the *reliability* that agent II has for agent Ω :

$$R^t(II, \Omega) = \sum_j P^t(o_j) \times R^t(II, \Omega, o_j)$$

4 Negotiation

For illustration II 's communication language is restricted to the illocutions: Offer(\cdot), Accept(\cdot), Reject(\cdot) and Withdraw(\cdot). The simple strategies that we will describe all use the same world model function, J_s , that maintains the following two probability distributions as their world model:

- $\mathbb{P}^t(II\text{Acc}(II, \Omega, \nu, \delta))$ — the strength of belief that II has in the proposition that she should accept the proposal $\delta = (a, b)$ from agent Ω in satisfaction of need ν at time t , where a is II 's commitment and b is Ω 's commitment. $\mathbb{P}^t(II\text{Acc}(II, \Omega, \nu, \delta))$ is estimated from:
 1. $\mathbb{P}^t(\text{Satisfy}(II, \Omega, \nu, \delta))$ a subjective evaluation (the strength of belief that II has in the proposition that the expected outcome of accepting the proposal will satisfy some of her needs).
 2. $\mathbb{P}^t(\text{Fair}(\delta))$ an objective evaluation (the strength of belief that II has in the proposition that the proposal is a “fair deal” in the open market).
 3. $\mathbb{P}^t(II\text{CanDo}(a))$ an estimate of whether II will be able to meet her commitment a at contract execution time.

These three arrays of probabilities are estimated by importing relevant information, $[info]$.

- $\mathbb{P}^t(\Omega\text{Acc}(\beta, \alpha, \delta))$ — the strength of belief that II has in the proposition that Ω would accept the proposal δ from agent II at time t . Every time that Ω submits a proposal she is revealing information about what she is prepared to accept, and every time she rejects a proposal she is revealing information about what she is *not* prepared to accept. Eg: having received the stamped illocution $\text{Offer}(\Omega, II, \delta)_{(\Omega, II, u)}$, at time $t > u$, II may believe that $\mathbb{P}^t(\Omega\text{Acc}(\Omega, II, \delta)) = \kappa$ this is used as a constraint on $\mathbb{P}^{t+1}(\Omega\text{Acc}(\cdot))$.

5 Virtual Institutions

This work is done on collaboration with the Spanish Governments IIIA Laboratory in Barcelona. Electronic Institutions are software systems composed of autonomous agents, that interact according to predefined conventions on language and protocol and that guarantee that certain norms of behaviour are enforced. Virtual Institutions enable rich interaction, based on natural language and embodiment of humans and software agents in a “liveable” vibrant environment. This view permits agents to behave autonomously and take their decisions freely up to the limits imposed by the set of *norms* of the institution. An important consequence of embedding agents in a virtual institution is that the predefined conventions on language and protocol greatly simplify the design of the agents. A Virtual Institution is in a sense a natural extension of the social concept of institutions as regulatory systems that shape human interactions [12].

Virtual Institutions are electronic environments designed to meet the following requirements towards their inhabitants:

- enable institutional commitments including structured language and norms of behaviour which enable reliable interaction between autonomous agents and between human and autonomous agents;
- enable rich interaction, based on natural language and embodiment of humans and software agents in a “liveable” vibrant environment.

The first requirement has been addressed to some extent by the Electronic Institutions (EI) methodology and technology for multi-agent systems, developed in the Spanish Government’s IIIA Laboratory in Barcelona [12]. The EI environment is oriented towards the engineering of multiagent systems. The Electronic Institution is an environment populated by autonomous software agents that interact according to predefined conventions on language and protocol. Following the metaphor of social institutions, Electronic Institutions guarantee that certain norms of behaviour are enforced. This view permits that agents behave autonomously and make their decisions freely up to the limits imposed by the set of norms of the institution. The interaction in such environment is regulated for software agents. The human, however, is “excluded” from the electronic institution.

The second requirement is supported to some extent by the distributed 3D Virtual Worlds technology. Emulating and extending the physical world in which we live, Virtual Worlds offer rich environment for a variety of human activities and multi-mode interaction. Both humans and software agents are embedded and visualised in such 3D environments as avatars, through which they communicate. The inhabitants of virtual worlds are aware of where they are and who is there — elements of the presence that are excluded from the current paradigm of e-Commerce environments. Following the metaphor of the physical world, these environments do not impose any regulations (in terms of language) on the interactions and any restrictions (in terms of norms of behaviour). When this encourages the social aspect of interactions and establishment of networks,

these environments do not provide means for enabling some behavioural norms, for example, fulfilling commitments, penalisation for misbehaviour and others.

Virtual Institutions addressed both requirements, retaining the features and advantages of the above discussed approaches. They can be seen as the logical evolution and merger of the two streams of development of environments that can host electronic markets as mixed societies of humans and software agents.

Technologically, Virtual Institutions are implemented following a three-layered framework, which provides deep integration of Electronic Institution technology and Virtual Worlds technology. The Electronic Institution Layer hosts the environments that support the Electronic Institutions technological component: the graphical EI specification designer ISLANDER and the runtime component AMELI [13]. At runtime, the Electronic Institution layer loads the institution specification and mediates agents interactions while enforcing institutional rules and norms.

The Communication Layer connects causally the Electronic Institutions layer with the 3D representation of the institution, which resides in the Social layer. The causal connection is the integrator. It enables the Electronic Institution layer to respond to changes in the 3D representation (for example, to respond to the human activities there), and passes back the response of the Electronic Institution layer in order to modify the corresponding 3D environment and maintain the consistency of the Virtual Institution. Virtual Institution representation is a graph and its topology can structure the space of the virtual environment in different ways. This is the responsibility of the Social layer. In this implementation the layer is represented in terms of a 3D Virtual World technology, structured around rooms, avatars, doors (for transitions) and other graphical elements. Technically, the Social layer is currently utilising Adobe Atmosphere virtual world technology. The design of the 3D World of the Virtual Institution is developed with the Annotation Editor, which ideally should take as an input a specification of the Electronic Institution layer and produce an initial layout of the 3D space. Currently, part of the work is done manually by a designer.

The core technology — the Causal Connection Server, enables the Communication Layer to act in two directions. Technically, in direction from the Electronic Institution layer, messages uttered by an agent have immediate impact in the Social layer. Transition of the agent between scenes in the Electronic Institution layer, for example, must let the corresponding avatar move within the Virtual World space accordingly. In the other direction, events caused by the actions of the human avatar in the Virtual World are transferred to the Electronic Institution layer and passed to an agent. This implies that actions forbidden to the agent by the norms of the institution (encoded in the Electronic Institution layer), cannot be performed by the human. For example, if a human needs to register first before leaving for the auction space, the corresponding agent is not allowed to leave the registration scene. Consequently, the avatar is not permitted to open the corresponding door to the auction.

6 Conclusions

A demonstrable prototype e-Market system permits both human and software agents to trade with each other on the World Wide Web. The main contributions described are: the broadly-based and “focussed” data mining systems, the intelligent agent architecture founded on information theory, and the abstract synthesis of the virtual worlds and the electronic institutions paradigms to form “virtual institutions”. These three technologies combine to present our vision of the World Wide Web marketplaces of tomorrow.

References

1. Lawrence, E., Newton, S., Corbitt, B., Lawrence, J., Dann, S., Thanasankit, T.: Internet Commerce — Digital Models for Business, 3rd edn. John Wiley and Sons, Inc., Chichester (2003)
2. Debenham, J., Simoff, S.: An e-Market Framework for Informed Trading. In: Carr, L., Roure, D.D., Iyengar, A., Goble, C., Dahlin, M. (eds.) proceedings 15th International World Wide Web Conference, WWW-2006. Edinburgh, Scotland (2006)
3. Debenham, J.: Bargaining with information. In: Jennings, N., Sierra, C., Sonenberg, L., Tambe, M. (eds.) Proceedings Third International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2004, pp. 664–671. ACM Press, New York (2004)
4. Sierra, C., Debenham, J.: Information-based agency. In: Proceedings of Twentieth International Joint Conference on Artificial Intelligence IJCAI 2007, Hyderabad, India, pp. 1513–1518 (2007)
5. Zhang, D., Simoff, S.: Informing the Curious Negotiator: Automatic news extraction from the Internet. In: Proceedings 3rd Australasian Data Mining Conference, Cairns, Australia, pp. 55–72 (2004)
6. Reis, D., Golgher, P.B., Silva, A., Laender, A.: Automatic web news extraction using tree edit distance. In: Proceedings of the 13th International Conference on the World Wide Web, New York, pp. 502–511 (2004)
7. Ramoni, M., Sebastiani, P.: Bayesian methods. In: Intelligent Data Analysis, pp. 132–168. Springer, Heidelberg (2003)
8. Zhang, D., Simoff, S., Debenham, J.: Exchange rate modelling using news articles and economic data. In: Proceedings of The 18th Australian Joint Conference on Artificial Intelligence, Sydney, Australia. Springer, Heidelberg (2005)
9. Faratin, P., Sierra, C., Jennings, N.: Using similarity criteria to make issue trade-offs in automated negotiation. *Journal of Artificial Intelligence* 142, 205–237 (2003)
10. Debenham, J.: Auctions and bidding with information. In: Faratin, P., Rodríguez-Aguilar, J.-A. (eds.) AMEC 2004. LNCS (LNAI), vol. 3435, pp. 15–28. Springer, Heidelberg (2006)
11. Debenham, J., Lawrence, E.: Intelligent agents that make informed decisions. In: Esposito, F., Raś, Z.W., Malerba, D., Semeraro, G. (eds.) ISMIS 2006. LNCS (LNAI), vol. 4203, pp. 137–146. Springer, Heidelberg (2006)
12. Arcos, J.L., Esteva, M., Noriega, P., Rodríguez, J.A., Sierra, C.: Environment engineering for multiagent systems. *Journal on Engineering Applications of Artificial Intelligence* 18 (2005)
13. EIDE (2005), <http://e-institutor.iiia.csic.es/>

Information Theory Based Intelligent Agents

Elaine Lawrence and John Debenham

QCIS, FEIT, University of Technology, Sydney, Australia
elaine@it.uts.edu.au

Abstract. Electronic and mobile trading environments are saturated with information from the Internet and the World Wide Web. This paper proposes agents that can assimilate and use real-time information flows wisely to automate contract negotiation reliably. A new breed of “information-based” agents are founded on concepts from information theory, and are designed to operate with information flows of varying and questionable integrity. These agents are part of a larger project that aims to make informed automated trading in applications such as electronic or mobile procurement a reality.

1 Introduction

Despite the substantial advances in multiagent systems and automated negotiation [1], it is perhaps surprising that negotiation in electronic business [2] remains a substantially manual procedure. Multi-agent systems often make use of human-agent teams especially in complex areas that concern contract negotiations. Electronic trading environments may be overwhelmed by information, including information drawn from general resources such as the World Wide Web using smart retrieval technology [3]. We propose that rather than strive to make strategic, economically rational decisions, intelligent agents in electronic markets should capitalise on the real-time information flows and should aim to make ‘informed decisions’ that take account of the integrity of all relevant information. Traditional agent architectures, such as the Belief-Desire-Intention (BDI) model, do not address directly the management of dynamic information flows of questionable integrity. This paper describes an agent architecture that has been designed specifically to operate in tandem with information discovery systems.

This is part of our e-Market Framework that is available on the World Wide Web [4]. This framework aims to make informed automated trading a reality, and aims to address the realities of electronic business, namely “what you know is the most important matter”. This work does not address all of the issues in automated trading [2]. For example, the work relies on developments in: XML and semantic web, secure data exchange, value chain management and financial services. Further the design of electronic marketplaces is not described here.

Intelligent agents and the information theory based architecture designed specifically to cope with real-time information flows are described in Sec. [2].

¹ <http://e-markets.org.au>

The management of dynamic information flows is described in Sec. 3. The interaction of more than one of these agents engaging in competitive negotiation is described in Sec. 4. Sec. 5 concludes.

2 Information-Theoretic Foundation for Agents

We have designed a new agent architecture founded on information theory. These “information-based” agents operate in real-time in response to market information flows. The central issues of trust in the execution of contracts is discussed in 4 5. The “information-based” agent’s reasoning is based on a first-order logic world model that manages multi-issue negotiation as easily as single-issue.

2.1 Rationale

This section provides the rationale for the formal work that follows.

Percepts, the content of messages, are all that an agent has to inform it about the world and other agents. The validity of percepts is constrained: by the agent’s uncertainty in the *reliability* of the sender of the message, by the *elapsed time* since the message arrived, and by the agent’s level of individual *caution in its belief*. The information-based agent’s *world model* is deduced from the percepts using *inference rules* that transform percepts into statements in probabilistic logic.

The integrity of percepts decreases in time. The way in which it decreases will be determined by the type of the percept, as well as by the issues such as uncertainty about a senders reliability, elapsed time and caution. An agent may have background knowledge concerning the expected integrity of a percept as $t \rightarrow \infty$. Information-based agents represent this background knowledge as a *decay limit distribution*. If the background knowledge is incomplete then one possibility for an agent is to assume that the decay limit distribution has maximum entropy whilst being consistent with the data.

All messages are valueless unless their integrity can be verified to some degree at a later time, perhaps for a cost. To deal with this issue we employ an *institution agent* that always reports promptly and honestly on the execution of all commitments, forecasts, promises and obligations. The institution agent is a simple solution to the integrity verification issue as well as determination of ownership of data. This enables the agents to negotiate and to evaluate the execution of commitments by simple message passing.

An agent’s percepts generally constitute a sparse data set whose elements have differing integrity. An agent may wish to induce tentative conclusions from this sparse and uncertain data of changing integrity. Percepts are transformed by inference rules into statements in probabilistic logic as described above. Information-based agents may employ entropy-based logic 6 to induce complete probability distributions from those statements. This logic is consistent with the laws of probability, but the results derived assume that the data is complete or, according to Watts Assumption, is ‘all that there is to know’.

An agent acts in response to some need or needs. A need may be exogenous such as the agents ‘owner’ needs to buy some brandy for example, or a message from another agent offering to trade may trigger a latent need to trade profitably. A need may also be endogenous such as the agent deciding that its owner has more brandy than is required. An agent may be attempting to satisfy a number of needs at any time, simultaneously and may have expectations of its future needs.

2.2 Agent Architecture

An agent observes events in its environment and represents some of those observations in its world model as beliefs. As time passes, an agent may not be prepared to accept such beliefs as being “true”, and qualifies those representations with epistemic probabilities. Those qualified representations of prior observations are the agents information. Given this information, an agent may then choose to adopt goals and strategies. Those strategies may be based on game theory, for example. To enable the agent’s strategies to make good use of its information, tools from information theory are applied to summarise and process that information. Such an agent is called *information-based*.

An agent called Π is the subject of this discussion. Π engages in multi-issue negotiation with a set of other agents: $\{\Omega_1, \dots, \Omega_o\}$, and information providing agents $\{\Theta_1, \dots, \Theta_k\}$. Π has two languages: \mathcal{C} and \mathcal{L} . \mathcal{C} is an illocutionary-based language for communication. \mathcal{L} is a first-order language for internal representation — precisely it is a first-order language with sentence probabilities optionally attached to each sentence representing Π ’s epistemic belief in the truth of that sentence. Messages expressed in \mathcal{C} from $\{\Theta_i\}$ and $\{\Omega_i\}$ are received, time-stamped, source-stamped and placed in an *in-box* \mathcal{X} . The messages in \mathcal{X} are then translated using an *import function* I into sentences expressed in \mathcal{L} that have integrity decay functions (usually of time) attached to each sentence, they are stored in a *repository* \mathcal{Y}^t . And that is all that happens until Π triggers a goal.

Π triggers a goal, $g \in \mathcal{G}$, in two ways: first in response to a message received from an opponent $\{\Omega_i\}$ “I offer you €100 in exchange for a pallet of paper”, and second in response to some need, $\nu \in \mathcal{N}$, “we need to order some more paper”. In either case, Π is motivated by a need — either a need to strike a deal, or a general need to trade. Π ’s goals could be short-term such as obtaining some information “what is the euro / dollar exchange rate?”, medium-term such as striking a deal with one of its opponents, or, rather longer-term such as building a (business) relationship with one of its opponents. So Π has a trigger mechanism T where: $T : \{\mathcal{X} \cup \mathcal{N}\} \rightarrow \mathcal{G}$.

For each goal that Π commits to, it has a mechanism, G , for selecting a strategy to achieve it where $G : \mathcal{G} \times \mathcal{M} \rightarrow \mathcal{S}$ where \mathcal{S} is the strategy library and \mathcal{M} the *world model*. A *strategy* s maps an information base into an action, $s(\mathcal{J}^t) = z \in \mathcal{Z}$. Given a goal, g , and the current state of the social model m^t , a strategy: $s = G(g, m^t)$. Each strategy, s , consists of a *plan*, b_s and a *world model* (construction and revision) *function*, J_s , that constructs, and maintains the currency of, the strategy’s *world model* W_s^t that consists of a set of probability

distributions. A *plan* derives the agent’s next action, z , on the basis of the agent’s world model for that strategy and the current state of the social model: $z = b_s(W_s^t, m^t)$, and $z = s(\mathcal{Y}^t)$. J_s employs two forms of entropy-based inference:

Maximum entropy inference. J_s^+ , first constructs an *information base* \mathcal{I}_s^t as a set of sentences expressed in \mathcal{L} derived from \mathcal{Y}^t , and then from \mathcal{I}_s^t constructs the world model, W_s^t , as a set of complete probability distributions [using Eqn. 2 in Sec. 2.3 below].

Maximum relative entropy inference. Given a prior world model, W_s^u , where $u < t$, minimum relative entropy inference, J_s^- , first constructs the incremental information base $\mathcal{I}_s^{(u,t)}$ of sentences derived from those in \mathcal{Y}^t that were received between time u and time t , and then from W_s^u and $\mathcal{I}_s^{(u,t)}$ constructs a new world model, W_s^t [using Eqn. 3 in Sec. 2.3 below].

2.3 Π ’s Reasoning

Once Π has selected a plan $a \in \mathcal{A}$ it uses maximum entropy inference to derive the $\{D_i^s\}_{i=1}^n$ and minimum relative entropy inference to update those distributions as new data becomes available. *Entropy*, \mathbb{H} , is a measure of uncertainty [7] in a probability distribution for a discrete random variable X : $\mathbb{H}(X) \triangleq -\sum_i p(x_i) \log p(x_i)$ where $p(x_i) = \mathbb{P}(X = x_i)$. Maximum entropy inference is used to derive sentence probabilities for that which is not known by constructing the “maximally noncommittal” [6] probability distribution, and is chosen for its ability to generate complete distributions from sparse data.

Let \mathcal{G} be the set of all positive ground literals that can be constructed using Π ’s language \mathcal{L} . A *possible world*, v , is a valuation function: $\mathcal{G} \rightarrow \{\top, \perp\}$. $\mathcal{V}|\mathcal{K}^s = \{v_i\}$ is the set of all possible worlds that are consistent with Π ’s knowledge base \mathcal{K}^s that contains statements which Π believes are true. A *random world* for \mathcal{K}^s , $W|\mathcal{K}^s = \{p_i\}$ is a probability distribution over $\mathcal{V}|\mathcal{K}^s = \{v_i\}$, where p_i expresses Π ’s degree of belief that each of the possible worlds, v_i , is the actual world. The *derived sentence probability* of any $\sigma \in \mathcal{L}$, with respect to a random world $W|\mathcal{K}^s$ is:

$$(\forall \sigma \in \mathcal{L}) \mathbb{P}_{\{W|\mathcal{K}^s\}}(\sigma) \triangleq \sum_n \{p_n : \sigma \text{ is } \top \text{ in } v_n\} \tag{1}$$

The agent’s *belief set* $\mathcal{B}_i^s = \{\Omega_j\}_{j=1}^M$ contains statements to which Π attaches a *given sentence probability* $\mathbb{B}(\cdot)$. A random world $W|\mathcal{K}^s$ is *consistent* with \mathcal{B}_i^s if: $(\forall \Omega \in \mathcal{B}_i^s)(\mathbb{B}(\Omega) = \mathbb{P}_{\{W|\mathcal{K}^s\}}(\Omega))$. Let $\{p_i\} = \{\overline{W}|\mathcal{K}^s, \mathcal{B}_i^s\}$ be the “maximum entropy probability distribution over $\mathcal{V}|\mathcal{K}^s$ that is consistent with \mathcal{B}_i^s ”. Given an agent with \mathcal{K}^s and \mathcal{B}_i^s , *maximum entropy inference* states that the *derived sentence probability* for any sentence, $\sigma \in \mathcal{L}$, is:

$$(\forall \sigma \in \mathcal{L}) \mathbb{P}_{\{\overline{W}|\mathcal{K}^s, \mathcal{B}_i^s\}}(\sigma) \triangleq \sum_n \{p_n : \sigma \text{ is } \top \text{ in } v_n\} \tag{2}$$

From Eqn. 2, each belief imposes a linear constraint on the $\{p_i\}$. The maximum entropy distribution: $\arg \max_{\underline{p}} \mathbb{H}(\underline{p})$, $\underline{p} = (p_1, \dots, p_N)$, subject to $M + 1$ linear

constraints: $g_j(\underline{p}) = \sum_{i=1}^N c_{ji} p_i - \mathbb{B}(\Omega_j) = 0, \quad j = 1, \dots, M. \quad g_0(\underline{p}) = \sum_{i=1}^N p_i - 1 = 0$, where $c_{ji} = 1$ if Ω_j is \top in v_i and 0 otherwise, and $p_i \geq 0, i = 1, \dots, N$, is found by introducing Lagrange multipliers, and then obtaining a numerical solution using the multivariate Newton-Raphson method. In the subsequent subsections we'll see how an agent updates the sentence probabilities depending on the type of information used in the update.

Given a prior probability distribution $\underline{q} = (q_i)_{i=1}^n$ and a set of constraints C , the *principle of minimum relative entropy* chooses the posterior probability distribution $\underline{p} = (p_i)_{i=1}^n$ that has the least *relative entropy*² with respect to \underline{q} :

$$\{\underline{W}|\underline{q}, C\} \triangleq \arg \min_{\underline{p}} \sum_{i=1}^n p_i \log \frac{p_i}{q_i}$$

and that satisfies the constraints. This may be found by introducing Lagrange multipliers as above. Given a prior distribution \underline{q} over $\{v_i\}$ — the set of all possible worlds, and a set of constraints C (that could have been derived as above from a set of new beliefs) *minimum relative entropy inference* states that the derived sentence probability for any sentence, $\sigma \in \mathcal{L}$, is:

$$(\forall \sigma \in \mathcal{L}) \mathbb{P}_{\{\underline{W}|\underline{q}, C\}}(\sigma) \triangleq \sum_n \{ p_n : \sigma \text{ is } \top \text{ in } v_n \} \tag{3}$$

where $\{p_i\} = \{\underline{W}|\underline{q}, C\}$. The principle of minimum relative entropy is a generalisation of the principle of maximum entropy. If the prior distribution \underline{q} is uniform, then the relative entropy of \underline{p} with respect to \underline{q} , $\underline{p}||\underline{q}$, differs from $-\mathbb{H}(\underline{p})$ only by a constant. So the principle of maximum entropy is equivalent to the principle of minimum relative entropy with a uniform prior distribution.

3 Managing Dynamic Information Flows

The illocutions in the communication language \mathcal{C} include information, $[info]$. The information received from general information sources will be expressed in terms defined by Π 's ontology. We define an *ontology signature* as a tuple $S = (C, R, \leq, \sigma)$ where C is a finite set of concept symbols (including basic data types); R is a finite set of relation symbols; \leq is a reflexive, transitive and anti-symmetric relation on C (a partial order); and, $\sigma : R \rightarrow C^+$ is the function assigning to each relation symbol its arity. Concepts play the role of *type*, and the is-a hierarchy is the notion of subtype. Thus, type inference mechanisms can be used to type all symbols appearing in expressions. We assume that Π makes at least part of that ontology public so that the other agents $\{\Omega_1, \dots, \Omega_o\}$ may communicate $[info]$ that Π can understand. Ω 's *reliability* is an estimate of the extent to which this $[info]$ is correct.

The only restriction on incoming $[info]$ is that it is expressed in terms of the ontology — this is very general. However, the way in which $[info]$ is used is

² Otherwise called *cross entropy* or the *Kullback-Leibler* distance between the two probability distributions.

completely specific — it will be represented as a set of linear constraints on one or more probability distributions in the world model. A chunk of $[info]$ may not be directly related to one of Π 's chosen distributions or may not be expressed naturally as constraints, and so some inference machinery is required to derive these constraints — this inference is performed by model building functions, J_s , that have been activated by a plan s chosen by Π . $J_s^D([info])$ denotes the set of constraints on distribution D derived by J_s from $[info]$.

3.1 Updating the World Model with $[info]$

The procedure for updating the world model as $[info]$ is received follows. If at time u , Π receives a message containing $[info]$ it is time-stamped and source-stamped $[info]_{(\Omega, \Pi, u)}$, and placed in a repository \mathcal{Y}^t . If Π has an active plan, s , with model building function, J_s , then J_s is applied to $[info]_{(\Omega, \Pi, u)}$ to derive constraints on some, or none, of Π 's distributions. The extent to which those constraints are permitted to effect the distributions is determined by a value for the *reliability* of Ω , $R^t(\Pi, \Omega, O([info]))$, where $O([info])$ is the ontological context of $[info]$.

An agent may have models of integrity decay for some particular distributions, but general models of integrity decay for, say, a chunk of information taken at random from the World Wide Web are generally unknown. However the values to which decaying integrity should tend in time *are* often known. For example, a prior value for the truth of the proposition that a “22 year-old male will default on credit card repayment” is well known to banks. If Π attaches such prior values to a distribution D they are called the *decay limit distribution* for D , $(d_i^D)_{i=1}^n$. No matter how integrity of $[info]$ decays, in the absence of any other relevant information it should decay to the decay limit distribution. If a distribution with n values has no decay limit distribution then integrity decays to the maximum entropy value $\frac{1}{n}$. In other words, the maximum entropy distribution is the default decay limit distribution.

In the absence of new $[info]$ the integrity of distributions decays. If $D = (q_i)_{i=1}^n$ then we use a geometric model of decay:

$$q_i^{t+1} = (1 - \rho^D) \times d_i^D + \rho^D \times q_i^t, \text{ for } i = 1, \dots, n \tag{4}$$

where $\rho^D \in (0, 1)$ is the decay rate. This raises the question of how to determine ρ^D . Just as an agent may know the decay limit distribution it may also know something about ρ^D . In the case of an information-overfed agent there is no harm in conservatively setting ρ^D “a bit on the low side” as the continually arriving $[info]$ will sustain the estimate for D .

We now describe how new $[info]$ is imported to the distributions. A single chunk of $[info]$ may effect a number of distributions. Suppose that a chunk of $[info]$ is received from Ω and that Π attaches the epistemic belief probability $R^t(\Pi, \Omega, O([info]))$ to it. Each distribution models a facet of the world. Given a distribution $D^t = (q_i^t)_{i=1}^n$, q_i^t is the probability that the possible world ω_i for D is the true world for D . The effect that a chunk $[info]$ has on distribution

D is to enforce the set of linear constraints on D , $J_s^D([info])$. If the constraints $J_s^D([info])$ are taken by Π as valid then Π could update D to the posterior distribution $(p_i^{[info]})_{i=1}^n$ that is the distribution with least relative entropy with respect to $(q_i^t)_{i=1}^n$ satisfying the constraint:

$$\sum_i \{p_i^{[info]} : J_s^D([info]) \text{ are all } \top \text{ in } \omega_i\} = 1. \tag{5}$$

But $R^t(\Pi, \Omega, O([info])) = r \in [0, 1]$ and Π should only treat the $J_s^D([info])$ as valid if $r = 1$. In general r determines the extent to which the effect of $[info]$ on D is closer to $(p_i^{[info]})_{i=1}^n$ or to the prior $(q_i^t)_{i=1}^n$ distribution by:

$$p_i^t = r \times p_i^{[info]} + (1 - r) \times q_i^t \tag{6}$$

But, we should only permit a new chunk of $[info]$ to influence D if doing so gives us new information. For example, if 5 minutes ago a trusted agent advises Π that the interest rate will go up by 1%, and 1 minute ago a very unreliable agent advises Π that the interest rate may go up by 0.5%, then the second unreliable chunk should not be permitted to ‘overwrite’ the first. We capture this by only permitting a new chunk of $[info]$ to be imported if the resulting distribution has more information *relative to* the decay limit distribution than the existing distribution has. Precisely, this is measured using the Kullback-Leibler distance measure — this is just one criterion for determining whether the $[info]$ should be used — and $[info]$ is only used if:

$$\sum_{i=1}^n p_i^t \log \frac{p_i^t}{d_i^D} > \sum_{i=1}^n q_i^t \log \frac{q_i^t}{d_i^D} \tag{7}$$

In addition, we have described in Eqn. 4 how the integrity of each distribution D will decay in time. Combining these two into one result, distribution D is revised to:

$$q_i^{t+1} = \begin{cases} (1 - \rho^D) \times d_i^D + \rho^D \times p_i^t & \text{if } [info] \text{ is usable} \\ (1 - \rho^D) \times d_i^D + \rho^D \times q_i^t & \text{otherwise} \end{cases}$$

for $i = 1, \dots, n$, and decay rate ρ^D as before.

3.2 Information Reliability

Sec. 3.1 relies on an estimate of $R^t(\Pi, \Omega, O([info]))$. This estimate is constructed by measuring the ‘error’ in observed information as the error in the effect that information has on each of Π ’s distributions. Suppose that a chunk of $[info]$ is received from agent Ω at time s and is verified at some later time t . For example, a chunk of information could be “the interest rate will rise by 0.5% next week”, and suppose that the interest rate actually rises by 0.25% — call that correct information $[fact]$. What does all this tell agent Π about agent Ω ’s reliability?

Consider one of Π 's distributions D that is $\{q_i^s\}$ at time s . Let $(p_i^{[info]})_{i=1}^n$ be the minimum relative entropy distribution given that $[info]$ has been received as calculated in Eqn. 5, and let $(p_i^{[fact]})_{i=1}^n$ be that distribution if $[fact]$ had been received instead. Suppose that the reliability estimate for distribution D was R_D^s . This section is concerned with what R_D^s should have been in the light of knowing *now*, at time t , that $[info]$ should have been $[fact]$, and how that knowledge effects our current reliability estimate for D , $R^t(\Pi, \Omega, O([info]))$.

The idea of Eqn. 6, is that the current value of r should be such that, *on average*, $(p_i^s)_{i=1}^n$ will be seen to be “close to” $(p_i^{[fact]})_{i=1}^n$ when we eventually discover $[fact]$ — no matter whether or not $[info]$ was used to update D , as determined by the acceptability test in Eqn. 7 at time s . That is, given $[info]$, $[fact]$ and the prior $(q_i^s)_{i=1}^n$, calculate $(p_i^{[info]})_{i=1}^n$ and $(p_i^{[fact]})_{i=1}^n$ using Eqn. 5. Then the *observed reliability* for distribution D , $R_D^{([info]||[fact])}$, on the basis of the verification of $[info]$ with $[fact]$ is the value of r that minimises the Kullback-Leibler distance between $(p_i^s)_{i=1}^n$ and $(p_i^{[fact]})_{i=1}^n$:

$$\arg \min_r \sum_{i=1}^n (r \cdot p_i^{[info]} + (1 - r) \cdot q_i^s) \log \frac{r \cdot p_i^{[info]} + (1 - r) \cdot q_i^s}{p_i^{[fact]}}$$

If $E^{[info]}$ is the set of distributions that $[info]$ affect, then the overall *observed reliability* on the basis of the verification of $[info]$ with $[fact]$ is: $R^{([info]||[fact])} = 1 - (\max_{D \in E^{[info]}} |1 - R_D^{([info]||[fact])}|)$. Then for each ontological context o_j , at time t when, perhaps, a chunk of $[info]$, with $O([info]) = o_k$, may have been verified with $[fact]$:

$$R^{t+1}(\Pi, \Omega, o_j) = (1 - \rho) \times R^t(\Pi, \Omega, o_j) + \rho \times R^{([info]||[fact])} \times \text{Sem}(o_j, o_k)$$

where $\text{Sem}(\cdot, \cdot) : O \times O \rightarrow [0, 1]$ measures the semantic distance [8] between two sections of the ontology, and ρ is the learning rate. Over time, Π notes the ontological context of the various chunks of $[info]$ received from Ω and over the various ontological contexts calculates the relative frequency, $P^t(o_j)$, of these contexts, $o_j = O([info])$. This leads to an overall expectation of the *reliability* that agent Π has for agent Ω : $R^t(\Pi, \Omega) = \sum_j P^t(o_j) \times R^t(\Pi, \Omega, o_j)$.

4 Negotiation

For illustration Π 's communication language [9] is restricted to the illocutions: Offer(\cdot), Accept(\cdot), Reject(\cdot) and Withdraw(\cdot). The simple strategies that we will describe all use the same world model function, J_s , that maintains the following two probability distributions as their world model:

- $\mathbb{P}^t(\Pi \text{Acc}(\Pi, \Omega, \nu, \delta))$ — the strength of belief that Π has in the proposition that she should accept the proposal $\delta = (a, b)$ from agent Ω in satisfaction of need ν at time t , where a is Π 's commitment and b is Ω 's commitment. $\mathbb{P}^t(\Pi \text{Acc}(\Pi, \Omega, \nu, \delta))$ is estimated from:

1. $\mathbb{P}^t(\text{Satisfy}(II, \Omega, \nu, \delta))$ a subjective evaluation (the strength of belief that II has in the proposition that the expected outcome of accepting the proposal will satisfy some of her needs).
2. $\mathbb{P}^t(\text{Fair}(\delta))$ an objective evaluation (the strength of belief that II has in the proposition that the proposal is a “fair deal” in the open market).
3. $\mathbb{P}^t(II\text{CanDo}(a))$ an estimate of whether II will be able to meet her commitment a at contract execution time.

These three arrays of probabilities are estimated by importing relevant information, [*info*], as described in Sec. 3.

- $\mathbb{P}^t(\Omega\text{Acc}(\beta, \alpha, \delta))$ — the strength of belief that II has in the proposition that Ω would accept the proposal δ from agent II at time t . Every time that Ω submits a proposal she is revealing information about what she is prepared to accept, and every time she rejects a proposal she is revealing information about what she is *not* prepared to accept. Eg: having received the stamped illocution $\text{Offer}(\Omega, II, \delta)_{(\Omega, II, u)}$, at time $t > u$, II may believe that $\mathbb{P}^t(\Omega\text{Acc}(\Omega, II, \delta)) = \kappa$ this is used as a constraint on $\mathbb{P}^{t+1}(\Omega\text{Acc}(\cdot))$ which is calculated using Eqn. 3.

4.1 Negotiation Strategies

An agent’s strategy s is a function of the information \mathcal{Y}^t that it has at time t . Four simple strategies make offers only on the basis of $\mathbb{P}^t(II\text{Acc}(II, \Omega, \nu, \delta))$, II ’s acceptability threshold γ , and $\mathbb{P}^t(\Omega\text{Acc}(\Omega, II, \delta))$. The greedy strategy s^+ chooses:

$$\arg \max_{\delta} \{ \mathbb{P}^t(II\text{Acc}(II, \Omega, \nu, \delta)) \mid \mathbb{P}^t(\Omega\text{Acc}(\Omega, II, \delta)) \gg 0 \}$$

it is appropriate when II believes Ω is desperate to trade.

The *expected-acceptability-to-II-optimizing strategy* s^* chooses:

$$\arg \max_{\delta} \{ \mathbb{P}^t(\Omega\text{Acc}(\Omega, II, \delta)) \times \mathbb{P}^t(II\text{Acc}(II, \Omega, \nu, \delta)) \mid \mathbb{P}^t(II\text{Acc}(II, \Omega, \nu, \delta)) \geq \gamma \}$$

when II is confident and not desperate to trade. The strategy s^- chooses:

$$\arg \max_{\delta} \{ \mathbb{P}^t(\Omega\text{Acc}(\Omega, II, \delta)) \mid \mathbb{P}^t(II\text{Acc}(II, \Omega, \nu, \delta)) \geq \gamma \}$$

it optimises the likelihood of trade — when II is keen to trade without compromising its own standards of acceptability.

An approach to issue-tradeoffs is described in [10]. The bargaining strategy described there attempts to make an acceptable offer by “walking round” the iso-curve of II ’s previous offer δ' (that has, say, an acceptability of $\gamma_{\delta'} \geq \gamma$) towards Ω ’s subsequent counter offer. In terms of the machinery described here, an analogue is to use the strategy s^- :

$$\arg \max_{\delta} \{ \mathbb{P}^t(\Omega\text{Acc}(\Omega, II, \delta)) \mid \mathbb{P}^t(II\text{Acc}(II, \Omega, \nu, \delta)) \geq \gamma_{\delta'} \}$$

with $\gamma = \gamma_{\delta'}$. This is reasonable for an agent that is attempting to be accommodating without compromising its own interests. The complexity of the strategy in [10] is linear with the number of issues. The strategy described here does not have that property, but it benefits from using $\mathbb{P}^t(\Omega\text{Acc}(\Omega, I, \delta))$ that contains foot prints of the prior offer sequence — estimated by repeated use of Eqn. 3 — in that distribution more recent data gives estimates with greater certainty.

5 Conclusions

This paper has described a new breed of “information-based” agents founded on concepts from information theory. We believe that to automate trading we must build intelligent agents that are ‘informed’, that can proactively acquire information to reduce uncertainty, that can estimate the integrity of real-time information flows, and can use uncertain information as a foundation for strategic decision-making [2]. An ‘information-based’ agent architecture has been described, that is founded on ideas from information theory, and has been developed specifically for this purpose.

References

1. Rosenschein, J.S., Zlotkin, G.: Rules of Encounter. The MIT Press, Cambridge (1994)
2. Lawrence, E., Newton, S., Corbitt, B., Lawrence, J., Dann, S., Thanasankit, T.: Internet Commerce — Digital Models for Business, 3rd edn. John Wiley and Sons, Inc., Chichester (2003)
3. Zhang, D., Simoff, S.: Informing the Curious Negotiator: Automatic news extraction from the Internet. In: Williams, G., Simoff, S. (eds.) Data Mining: Theory, Methodology, Techniques, and Applications, pp. 176–191. Springer, Heidelberg (2006)
4. Sierra, C., Debenham, J.: Information-based agency. In: Proceedings of Twentieth International Joint Conference on Artificial Intelligence IJCAI 2007, Hyderabad, India, pp. 1513–1518 (2007)
5. Debenham, J., Simoff, S.: An e-Market Framework for Informed Trading. In: Carr, L., Roure, D.D., Iyengar, A., Goble, C., Dahlin, M. (eds.) proceedings 15th International World Wide Web Conference, WWW-2006, Edinburgh, Scotland (2006)
6. Jaynes, E.: Probability Theory — The Logic of Science. Cambridge University Press, Cambridge (2003)
7. MacKay, D.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
8. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on Knowledge and Data Engineering 15, 871–882 (2003)
9. Rahwan, I., Ramchurn, S., Jennings, N., McBurney, P., Parsons, S., Sonenberg, E.: Argumentation-based negotiation. Knowledge Engineering Review 18, 343–375 (2003)
10. Debenham, J., Lawrence, E.: Intelligent agents that make informed decisions. In: Esposito, F., Raś, Z.W., Malerba, D., Semeraro, G. (eds.) ISMIS 2006. LNCS (LNAI), vol. 4203, pp. 137–146. Springer, Heidelberg (2006)

A Possibilistic Approach to Goal Generation in Cognitive Agents

Célia da Costa Pereira and Andrea G.B. Tettamanzi

Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
via Bramante 65, I-26013 Crema, Italy
{celia.pereira, andrea.tettamanzi}@unimi.it

Abstract. We propose a theoretical framework, grounded in possibility theory, to account for all the aspects involved in representing and changing beliefs, representing and generating justified desires, and selecting goals based on current beliefs about the world and the preferences of an agent.

1 Introduction and Related Work

There is a consensus among researchers that the generation of the goals to be adopted by an agent depends on its mental state [4,5,7]. It can be the result of an explicit external request that is *accepted* by the agent, e.g., [14]; or a *consequence* of the agent's mental attitudes, e.g., [5]. Instead, the choice of the *best* set of goals to be adopted (pursued) depends also on the *consistency* (or *feasibility*) of such goals.

There are two directions followed by researchers to define desire/goal consistency. One considers goal generation and adoption as a whole, the other considers them as two separate steps. In the former case, the evaluation of the consistency of a desire set takes the cognitive components into account, but can lead to a consistent but sub-optimal goal set. In the latter case, the evaluation of consistency does not take the cognitive components of the agent into account (logical consistency). This can lead the agent to choose sets of desires which are logically consistent but inconsistent from the cognitive point of view.

We propose a possibilistic approach in which the generation and the adoption parts are considered separately and propose a new and possibilistic-based definition of desire/goal consistency which incorporates the two points of view. Using a possibilistic framework to represent beliefs and desires allows us to also represent *partially sure beliefs* and *partially desirable* world states.

To make justice to the complexity of real world, the agent's beliefs are represented by a possibility distribution and we adapt the belief conditioning operator proposed by Dubois and Prade [9] to update the beliefs.

A consequence of representing beliefs as a matter of degree is that desires also have to be considered as such. Like beliefs, desires are represented by a possibility distribution that induces a complete preorder over the set of possible worlds. However, for the sake of simplicity, we make the assumption that an agent only generates positive desires [2].

2 Possibilistic Representation

The representation of beliefs and desires calls for a quick recall of possibility theory.

2.1 Possibility Theory

Possibility theory is a mathematical theory of uncertainty that relies upon fuzzy set theory [15], in that the (fuzzy) set of possible values for a variable of interest is used to describe the uncertainty as to its precise value. The membership function of such set, π , is called a *possibility distribution*.

A possibility distribution for which there exists a completely possible value ($\exists v_0; \pi(v_0) = 1$) is said to be *normalized*.

Definition 1 (Possibility and Necessity Measures). *A possibility distribution π induces a possibility measure and its dual necessity measure, denoted by Π and N respectively. Both measures apply to a crisp set A and are defined as follows:*

$$\Pi(A) = \max_{s \in A} \pi(s); \quad (1)$$

$$N(A) = 1 - \Pi(\bar{A}) = \min_{s \in \bar{A}} \{1 - \pi(s)\}. \quad (2)$$

Another interesting measure that can be defined based on a possibility distribution is *guaranteed possibility* [11].

Definition 2 (Guaranteed Possibility Measure)

Given a possibility distribution π , a guaranteed possibility measure, noted Δ , is defined as:

$$\Delta(A) = \min_{s \in A} \pi(s); \quad (3)$$

A few properties of possibility, necessity, and guaranteed possibility measures induced by a normalized possibility distribution on a finite universe of discourse Ω are the following. For all subsets $A, B \subseteq \Omega$:

1. $\Pi(A \cup B) = \max\{\Pi(A), \Pi(B)\}$, $N(A \cap B) = \min\{N(A), N(B)\}$;
2. $\Pi(\emptyset) = N(\emptyset) = 0$, $\Pi(\Omega) = N(\Omega) = 1$, $\Pi(A) = 1 - N(\bar{A})$;
3. $N(A) \leq \Pi(A)$, $\Delta(A) \leq \Pi(A)$;
4. $N(A) > 0$ implies $\Pi(A) = 1$, $\Pi(A) < 1$ implies $N(A) = 0$;

A consequence of these properties is that $\max\{\Pi(A), \Pi(\bar{A})\} = 1$. In case of complete ignorance on A , $\Pi(A) = \Pi(\bar{A}) = 1$.

2.2 Language and Interpretations

Information manipulated by a cognitive agent must be represented symbolically. To develop our theoretical framework, we adopt perhaps the simplest symbolic representation, in the form of a classical propositional language.

Definition 3 (Language). Let \mathcal{A} be a finite¹ set of atomic propositions and let \mathcal{L} be the propositional language such that $\mathcal{A} \cup \{\top, \perp\} \subseteq \mathcal{L}$, and, $\forall \phi, \psi \in \mathcal{L}$, $\neg\phi \in \mathcal{L}$, $\phi \wedge \psi \in \mathcal{L}$, $\phi \vee \psi \in \mathcal{L}$.

We will denote by $\Omega = \{0, 1\}^{\mathcal{A}}$ the set of all interpretations on \mathcal{A} . An interpretation $\mathcal{I} \in \Omega$ is a function $\mathcal{I} : \mathcal{A} \rightarrow \{0, 1\}$ assigning a truth value $p^{\mathcal{I}}$ to every atomic proposition $p \in \mathcal{A}$ and, by extension, a truth value $\phi^{\mathcal{I}}$ to all formulas $\phi \in \mathcal{L}$.

Definition 4. The notation $[\phi]$ denotes the set of all models (i.e., interpretations satisfying ϕ) of a formula $\phi \in \mathcal{L}$: $[\phi] = \{\mathcal{I} \in \Omega : \mathcal{I} \models \phi\}$. Likewise, if $S \subseteq \mathcal{L}$ is a set of formulas, $[S] = \{\mathcal{I} \in \Omega : \forall \phi \in S, \mathcal{I} \models \phi\} = \bigcap_{\phi \in S} [\phi]$.

2.3 Representing Beliefs and Desires

The beliefs and desires of a cognitive agent are represented thanks to two different possibility distributions π and u respectively. Thus, a normalized possibility distribution π means that there exists at least one possible situation which is consistent with the available knowledge. All these considerations are in line with what is proposed in [2,10] with the following differences: (i) the qualitative utilities associated to positive desires are the result of a deliberative process, that is, they depend on the mental state of the agent; (ii) desires may be inconsistent and a way to calculate the degree of (logical and cognitive) consistency of the agent’s desires is proposed; (iii) there is an explicit distinction between *goals* and *desires*; (iv) for the sake of simplicity we do not consider negative desires.

Representing Desires. The desires of an agent depend on its beliefs. While desires are represented by means of a possibility distribution, one must understand that such a distribution is just an epiphenomenon of an underlying, more primitive mechanism which determines how desires arise. A description of such mechanism is given in terms of desire-generation rules.

Definition 5 (Desire-Generation Rule). A desire-generation rule R is an expression of the form $\beta_R, \psi_R \Rightarrow_D^+ \phi$, where $\beta_R, \psi_R, \phi \in \mathcal{L}$. The unconditional counterpart of this rule is $\alpha \Rightarrow_D^+ \phi$, with $\alpha \in (0, 1]$.

The intended meaning of a conditional desire-generation rule is: “an agent desires every world in which ϕ is true at least as much as it believes β_R and desires ψ_R ”, or, put in terms of qualitative utility, “the qualitative utility attached by the agent to every world satisfying ϕ is greater than, or equal to, the degree to which it believes β_R and desires ψ_R ”. The intended meaning of an unconditional rule is that the qualitative utility of every world $\mathcal{I} \models \phi$ is at least α for the agent.

Given a desire-generation rule R , we shall denote $\text{rhs}(R)$ the formula on the right-hand side of R .

¹ Like in [3], we adopt the restriction to the finite case in order to use standard definitions of possibilistic logic. Extensions to the infinite case are discussed, for example, in [8].

Representing Graded Beliefs. A *belief*, which is a component of an agent's cognitive state, can be regarded as a necessity degree induced by a normalized possibility distribution $\pi : \Omega \rightarrow [0, 1]$ on the possible worlds. The possibility degree $\pi(\mathcal{I})$ represents the plausibility order of the possible world situation represented by interpretation \mathcal{I} .

Definition 6 (Graded Belief). Let N be the necessity measure induced by π , and ϕ be a formula. The degree to which the agent believes ϕ is given by:

$$\mathcal{B}(\phi) = N([\phi]) = 1 - \max_{\mathcal{I} \not\models \phi} \{\pi(\mathcal{I})\}. \quad (4)$$

Straightforward consequences of the properties of possibility and necessity measures are that $\mathcal{B}(\phi) > 0 \Rightarrow \mathcal{B}(\neg\phi) = 0$, this means that if the agent somehow believes ϕ then it cannot believe $\neg\phi$ at all; $\mathcal{B}(\top) = 1$, $\mathcal{B}(\perp) = 0$, and

$$\mathcal{B}(\phi \wedge \psi) = \min\{\mathcal{B}(\phi), \mathcal{B}(\psi)\}, \quad \mathcal{B}(\phi \vee \psi) \geq \max\{\mathcal{B}(\phi), \mathcal{B}(\psi)\}. \quad (5)$$

2.4 Mental State

We now have the elements to define the mental state of an agent, which consists of its beliefs and the rules defining the deliberation mechanism whereby desires are generated based on beliefs.

Definition 7 (Mental State). The state of an agent is completely described by a pair $S = \langle \pi, \mathcal{R}_J \rangle$, where

- π is a possibility distribution which induces the agent's beliefs \mathcal{B} ;
- \mathcal{R}_J is a set of desire-generation rules which, together with \mathcal{B} , induce a qualitative utility assignment u .

Example. Dr. A. Gent has submitted a paper to IEAAIE 2010 he has written with his co-author I. M. Flaky, who has promised to go to Córdoba to present it if it is accepted. Dr. Gent knows that, if the paper is accepted, publishing it (which is his great desire), means to pay the conference registration (for his co-author or for himself) and then be ready to go to Córdoba to present it, in case I. M. is unavailable.

If the paper is accepted (a), Dr. Gent is willing to pay the registration (r); furthermore, if the paper is accepted and Dr. Flaky turns out to be unavailable (q), he is willing to go to Córdoba to present it (p). Finally, if he knows the paper is accepted and wishes to present it, he will desire to have a hotel room (h) and a plane ticket reserved (t).

Dr. Gent has some *a priori* beliefs about this situation, namely that if the hotels are all booked out (b), he will not succeed in booking a hotel room; similarly, he believes that if the planes are full (f), he will not succeed in reserving a flight, although this is not necessarily true, if he puts himself in the waiting list and a reservation is cancelled. Finally, he believes the organizers will enforce the rule whereby his paper will be presented only if it is accepted and a registration is paid.

The set of atomic propositions is then $\mathcal{A} = \{a, b, f, h, p, r, t, q\}$ and

$$\mathcal{R}_J = \left\{ \begin{array}{l} R_1 : a, p \Rightarrow \frac{+}{D} t \wedge h, \\ R_2 : a \wedge q, \top \Rightarrow \frac{+}{D} p, \\ R_3 : a, \top \Rightarrow \frac{+}{D} r. \end{array} \right\}.$$

3 Beliefs

The belief change operator used here is an adaptation of Dubois and Prade's belief conditioning operator [9] and allows to update the possibility distribution π in light of new trusted information.

A source of information is considered trusted to a certain extent. This means that its membership degree to the fuzzy set of trusted sources is a value $\tau \in [0, 1]$. Let $\phi \in \mathcal{L}$ be incoming information from a source trusted to degree τ . The belief change operator is defined as follows:

Definition 8 (Belief Change Operator). *The possibility distribution π' which induces the new belief set \mathcal{B}' after receiving information ϕ is computed from possibility distribution π relevant to the previous belief set \mathcal{B} ($\mathcal{B}' = \mathcal{B} * \frac{\tau}{\phi}$, $\pi' = \pi * \frac{\tau}{\phi}$) as follows: for all interpretation \mathcal{I} ,*

$$\pi'(\mathcal{I}) = \begin{cases} \frac{\pi(\mathcal{I})}{\Pi(\{\phi\})}, & \text{if } \mathcal{I} \models \phi \text{ and } \mathcal{B}(\neg\phi) < 1; \\ 1, & \text{if } \mathcal{I} \models \phi \text{ and } \mathcal{B}(\neg\phi) = 1; \\ \min\{\pi(\mathcal{I}), (1 - \tau)\}, & \text{if } \mathcal{I} \not\models \phi. \end{cases} \quad (6)$$

The second case in Equation 6 provides for the *revision* of beliefs that contradict ϕ . In general, the operator treats new information ϕ in the negative sense: being told ϕ denies the possibility of world situations where ϕ is false (third case of Equation 6). The possibility of world situations where ϕ is true may only increase due to the first case in equation 6 or revision (second case of Equation 6). If information from a fully trusted source contradicts an existing proposition that is fully believed, then revising with the above operator leads the agent to believe the more recent information and give up the oldest to restore consistency. Finally, it can be shown that the belief change operator $*$ obeys a possibilistic version of the AGM revision rationality postulates [12]. It is easy to verify that the $*$ operator is a generalization of the possibilistic conditioning operator of Dubois and colleagues [9].

Example (continued). Dr. Gent's *a priori* beliefs may be modeled by assuming Dr. Gent at some point had no beliefs at all (π is 1 everywhere), and then was "told":

- $f \supset \neg h$ with certainty 1 (i.e., $f \supset \neg h$ by a fully trusted source),
- $b \supset \neg t$ with certainty 0.9 (i.e., $b \supset \neg t$ by a source with trust $\tau = 0.9$),
- $\neg(r \wedge a) \supset \neg p$ with certainty 1,

which yields the possibility distribution shown in Figure 11

At this point, the following happens:

- $1/a$: Dr. Gent receives the notification of acceptance of his paper: the source is the program chair of IEAAIE, whom Dr. Gent trusts in full;
- $0.75/q$: soon after learning that the paper has been accepted, Dr. Flaky rushes into Dr. Gent's office to inform him that he is no more available to go to Córdoba; as always, Dr. Gent does not completely trust what Dr. Flaky tells him, as he is well-known for changing his mind very often;

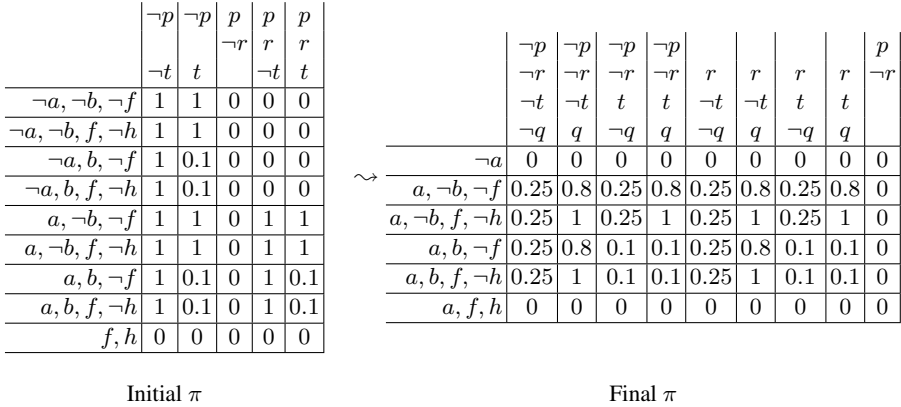


Fig. 1. Dr. Gent’s initial and final possibility distribution. Interpretations have been grouped together where possible, due to lack of space: when no literal appears for a given atom in a row or column heading, it is understood that the row or column applies for both truth assignments.

- 0.2/f: a few weeks later, Dr. Gent meets a colleague who tells him he has heard another colleague say someone on IEAAIE’s organizing committee told her all the hotel rooms in Córdoba are already booked out; Dr. Gent considers this news as yet unverified; nevertheless, he takes notice of it.

Dr. Gent’s beliefs are represented by the “final” possibility distribution shown in Figure 1.

4 Desires

We suppose that the agent’s subjective qualitative utilities are determined dynamically through a rule-based deliberation mechanism. Associating a qualitative utility first to worlds and not to formulas allows us to (i) directly construct the possibility distribution u ; and (ii) makes it possible to also calculate the qualitative degree of formulas which do not appear explicitly on the right-hand side of any rule.

Like in [3], the qualitative utility associated to each positive desire formula is computed on the basis of the guaranteed possibility measure Δ .

The set of the agent’s justified positive desires, \mathcal{J} , is induced by the assignment of a qualitative utility u , which, unlike π , needs not be normalized, since desires may very well be inconsistent.

Definition 9 (Justified Desire). *Given a qualitative utility assignment u (formally a possibility distribution), the degree to which the agent desires $\phi \in \mathcal{L}$ is given by*

$$\mathcal{J}(\phi) = \Delta([\phi]) = \min_{\mathcal{I} \models \phi} u(\mathcal{I}). \tag{7}$$

In words, the degree of justification of a desire is given by the guaranteed qualitative utility of the set of all worlds in which the desire would be fulfilled. Intuitively, a desire is justified to the extent that all the worlds in which it is fulfilled are desirable.

Interpreting $\mathcal{J}(\phi)$ as a degree of membership defines the fuzzy set \mathcal{J} of the agent's justified positive desires.

In turn, a qualitative utility assignment u is univocally determined by the mental state of the agent as explained below.

Definition 10 (Rule Activation). Let $R = \beta_R, \psi_R \Rightarrow_D^+ \phi$ be a desire-generation rule. The degree of activation of R , $\text{Deg}(R)$, is given by

$$\text{Deg}(R) = \min\{\mathcal{B}(\beta_R), \mathcal{J}(\psi_R)\}.$$

For an unconditional rule $R = \alpha_R \Rightarrow_D^+ \phi$, $\text{Deg}(R) = \alpha_R$.

Let us denote by $\mathcal{R}_J^\mathcal{I} = \{R \in \mathcal{R}_J : \mathcal{I} \models \text{rhs}(R)\}$ the subset of \mathcal{R}_J containing just the rules whose right-hand side would be true in world \mathcal{I} .

Definition 11 (Desired Worlds). The qualitative utility assignment $u : \Omega \rightarrow [0, 1]$ (formally a possibility distribution) is defined, for all $\mathcal{I} \in \Omega$, as

$$u(\mathcal{I}) = \max_{R \in \mathcal{R}_J^\mathcal{I}} \text{Deg}(R). \tag{8}$$

The apparent circularity of Definitions 9, 10, and 11 is resolved by an algorithmic translation which reveals u is the limit distribution obtained by iteratively applying the definitions. Given a mental state $\mathcal{S} = \langle \pi, \mathcal{R}_J \rangle$, the corresponding qualitative utility assignment, u , is computed by the following algorithm.

Algorithm 1 (Deliberation)

1. $i \leftarrow 0$; for all $\mathcal{I} \in \Omega$, $u_0(\mathcal{I}) \leftarrow 0$;
2. $i \leftarrow i + 1$;
3. For all $\mathcal{I} \in \Omega$,

$$u_i(\mathcal{I}) \leftarrow \begin{cases} \max_{R \in \mathcal{R}_J^\mathcal{I}} \text{Deg}_{i-1}(R), & \text{if } \mathcal{R}_J^\mathcal{I} \neq \emptyset, \\ 0, & \text{otherwise,} \end{cases}$$

where $\text{Deg}_{i-1}(R)$ is the degree of activation of rule R calculated using u_{i-1} as the qualitative utility assignment;

4. if $\max_{\mathcal{I}} |u_i(\mathcal{I}) - u_{i-1}(\mathcal{I})| > 0$, i.e., if a fixpoint has not been reached yet, go back to Step 2;
5. For all $\mathcal{I} \in \Omega$, $u(\mathcal{I}) \leftarrow u_i(\mathcal{I})$; u is the qualitative utility assignment corresponding to mental state \mathcal{S} .

Proposition 1. Algorithm 1 always terminates.

Example (continued). Dr. Gent's desires may now be determined, based on the desire-generation rules R_1 , R_2 , and R_3 , and Dr. Gent's beliefs, represented by the possibility distribution shown in Figure 1 by applying Algorithm 1 which stops at iteration $i = 3$ with the qualitative utility distribution shown in Figure 2.

As expected, $\mathcal{J}(r) = 1$ and $\mathcal{J}(t \wedge h) = \mathcal{J}(p) = 0.75$, but $\mathcal{J}(t) = \mathcal{J}(h) = 0$. However, these are not all of Dr. Gent's desires. Other desires are justified under these conditions, for instance $\mathcal{J}(\neg a \wedge p) = 0.75$ and $\mathcal{J}(\neg a \wedge r) = 1$, and even $\mathcal{J}(\neg r \wedge p) = 0,75$ and $\mathcal{J}(r \wedge \neg p) = 1$.

| | | | | | |
|----------|----------|----------|----------|----------|-----|
| | $\neg p$ | $\neg p$ | $\neg p$ | p | p |
| | $\neg r$ | $\neg r$ | r | $\neg r$ | r |
| | $\neg t$ | t | | | |
| $\neg h$ | 0 | 0 | 1 | 0.75 | 1 |
| h | 0 | 0.75 | 1 | 0.75 | 1 |

Fig. 2. Dr. Gent’s final qualitative utility distribution

5 Goals

Here, we make a clear distinction between desires and goals. Desires may be inconsistent. Goals, instead, are defined as a consistent subset of desires.

Definition 12. *The overall possibility of a set $S \subseteq \mathcal{L}$ of formulas is*

$$\Pi([S]) = \max_{\mathcal{I} \in [S]} \pi(\mathcal{I}). \tag{9}$$

The following definition extends \mathcal{J} , the degree of justification of a desire, to sets of desires.

Definition 13. *The overall justification of a set $S \subseteq \mathcal{L}$ of formulas is*

$$\mathcal{J}(S) = \Delta([S]) = \min_{\mathcal{I} \in [S]} u(\mathcal{I}). \tag{10}$$

Therefore, by the properties of the minimum guaranteed possibility:

Proposition 2. *The justified degree of a set of desires is greater than or equal to all the justified degree of each desire in the considered set, formally:*

$$\mathcal{J}(S) \geq \max_{\phi \in S} \{\mathcal{J}(\phi)\}. \tag{11}$$

Proposition 3. *The addition of a desire to a set of desires cannot lead to a decrease of the justification level of the resulting enlarged set of desires. Let $S \subseteq \mathcal{L}$ be a set of desires. For all desire ϕ ,*

$$\mathcal{J}(S \cup \{\phi\}) \geq \mathcal{J}(S); \tag{12}$$

$$\mathcal{J}(S) \geq \mathcal{J}(S \setminus \{\phi\}). \tag{13}$$

A rational agent will select as goals the set of desires that, besides being logically “consistent”, is also maximally desirable, i.e., maximally justified. The problem with logical “consistency”, however, is that it does not capture “implicit” inconsistencies among desires, that is consistency due to the agent beliefs (I adopt as goals only desires which are not inconsistent with my beliefs). Therefore, a suitable definition of desire consistency in the possibilistic setting is required. Such definition must take the agent’s cognitive state into account as pointed out, for example, in [6, 13, 11].

For example, an agent desires p and desires q , believing that $p \supset \neg q$. Although $\{p, q\}$, as a set of formulas, i.e., syntactically, is logically consistent, it is not if one take the belief $p \supset \neg q$ into account.

We argue that a suitable definition of such “cognitive” consistency is one based on the possibility of the set of desires, as defined above. Indeed,

Definition 14. *a set of desires S is consistent, in the cognitive sense, if and only if $\Pi([S]) > 0$.*

Of course, this definition of cognitive consistency implies logical consistency: if S is logically inconsistent, $\Pi([S]) = 0$. We will take a step forward, by assuming a rational agent will select as goals the most desirable set of desires among the most possible such sets.

Let $\mathcal{D} = \{S \subseteq \text{supp}(\mathcal{J})\}$, i.e., the set of desire sets whose justification is greater than zero. Given $\gamma \in (0, 1]$, $\mathcal{D}_\gamma = \{S \in \mathcal{D} : \Pi([S]) \geq \gamma\}$ is the subset of \mathcal{D} containing only those sets whose overall possibility is at least γ .

For every given level of possibility γ , a rational agent will elect as its goal set the maximally desirable of the γ -possible sets.

Definition 15 (Goal set). *The γ -possible goal set is*

$$G_\gamma = \begin{cases} \arg \max_{S \in \mathcal{D}_\gamma} \mathcal{J}(S) & \text{if } \mathcal{D}_\gamma \neq \emptyset, \\ \emptyset & \text{otherwise.} \end{cases}$$

We denote by γ^* the maximum possibility level such that $G_\gamma \neq \emptyset$. Then, the goal set elected by a rational agent will be

$$G^* = G_{\gamma^*}, \quad \gamma^* = \max_{G_\gamma \neq \emptyset} \gamma. \tag{14}$$

Proposition 4. *A rational agent chooses a goal set that is maximally possible (with a non zero possibility anyway) and maximally justifiable.*

Example (continued). To determine a consistent set of goals to commit to, Dr. Gent must perform a goal election, which, in this case, yields $\gamma^* = 1$ and $G^* = \{r\}$: Dr. Gent must pay the registration, that is for sure; planning his trip is less urgent, for Dr. Flaky, as far as Dr. Gent believes, might still change his mind.

6 Conclusion

A theoretical framework for goal generation in BDI agent has been justified and developed. Beliefs and desires are represented by means of two possibility distributions. A deliberative process is responsible for generating the distribution of qualitative utility that underlies desire justification, and the election of goals considers their cognitive consistency, realized as possibility. Due to lack of space, all the proofs have been omitted.

² Let A be a fuzzy set, $\text{supp}(A)$, is the set of all x such that $A(x) > 0$.

References

1. Baker, D.: Ambivalent desires and the problem with reduction. *Philosophical Studies* (Published online: March 25, 2009)
2. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Bipolar possibility theory in preference modeling: Representation, fusion and optimal solutions. *Inf. Fusion* 7(1), 135–150 (2006)
3. Benferhat, S., Kaci, S.: Logical representation and fusion of prioritized information based on guaranteed possibility measures: application to the distance-based merging of classical bases. *Artif. Intell.* 148(1-2), 291–333 (2003)
4. Broersen, J., Dastani, M., Hulstijn, J., van der Torre, L.: Goal generation in the BOID architecture. *Cognitive Science Quarterly Journal* 2(3–4), 428–447 (2002)
5. Castelfranchi, C., Paglieri, F.: The role of beliefs in goal dynamics: Prolegomena to a constructive theory of intentions. *Synthese* 155(2), 237–263 (2007)
6. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. *Artif. Intell.* 42(2-3), 213–261 (1990)
7. da Costa Pereira, C., Tettamanzi, A.: Goal generation and adoption from partially trusted beliefs. In: *Proceedings of ECAI 2008*, pp. 453–457. IOS Press, Amsterdam (2008)
8. De Baets, B., Tsiporkova, E., Mesiar, R.: Conditioning in possibility theory with strict order norms. *Fuzzy Sets Syst.* 106(2), 221–229 (1999)
9. Dubois, D., Prade, H.: A synthetic view of belief revision with uncertain inputs in the framework of possibility theory. *International Journal of Approximate Reasoning* 17, 295–324 (1997)
10. Dubois, D., Prade, H.: An introduction to bipolar representations of information and preference. *Int. J. Intell. Syst.* 23(8), 866–877 (2008)
11. Dubois, D., Prade, H.: An overview of the asymmetric bipolar representation of positive and negative information in possibility theory. *Fuzzy Sets Syst.* 160(10), 1355–1366 (2009)
12. Gärdenfors, P.: Belief revision: A vademecum. In: *Meta-Programming in Logic*, pp. 1–10. Springer, Berlin (1992)
13. Rao, A.S., Georgeff, M.P.: Asymmetry thesis and side-effect problems in linear-time and branching-time intention logics. In: *IJCAI*, pp. 498–505 (1991)
14. Shapiro, S., Lespérance, Y., Levesque, H.J.: Goal change. In: *Proceedings of IJCAI 2005*, pp. 582–588 (2005)
15. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8, 338–353 (1965)

Modelling Greed of Agents in Economical Context

Tibor Bosse¹, Ghazanfar F. Siddiqui^{1,2}, and Jan Treur¹

¹ Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

² Quaid-i-Azam University Islamabad, Department of Computer Science, 45320, Pakistan
{tbosse, ghazanfa, treur}@few.vu.nl, ghazanfar@qau.edu.pk
<http://www.few.vu.nl/~{tbosse, ghazanfa, treur}>

Abstract. A classical debate in economics addresses the advantages and drawbacks of modelling from a macroeconomics perspective as opposed to modelling from a microeconomics perspective. Form the latter psychological aspects at an individual level can be taken into account in a differentiated manner. Within computer science and AI, a similar debate exists about the differences between agent-based and population-based modelling. This paper aligns both debates by exploring the differences and commonalities between population-based and agent-based modelling in economical context. A case study is performed on the interplay between individual greed as a psychological concept and global economical concepts. It is shown that under certain conditions agent-based and population-based simulations show similar results.

Keywords: economics, greed, agent-based and population-based modelling.

1 Introduction

Traditionally, macroeconomics addresses the behaviour of a world-wide, national or regional economy as a whole [3], whereas microeconomics investigates the economic behaviour and decision making of individual agents, for example, consumers, households or firms [11]. Since the latter aims to understand why and how agents make certain economic decisions, various social, cognitive, and emotional factors of human behaviour are studied. This has resulted in the emergence of the field of behavioural economics [14]. Although this may be very useful when one wants to analyse the behaviour of individual agents, there is some debate about the extent to which it is useful to incorporate these aspects when studying global processes in economics, e.g., [5]. Do personal factors such as risk avoidance, greed, and personal circumstances provide more insight in the global patterns, or can they simply be ignored or treated in a more abstract, aggregated manner? This paper provides some answers to these questions from a computational perspective.

In recent years, various authors have studied processes in economics by building computational models of them, and analysing the dynamics of these models using agent-based simulation techniques [15]. Ironically, also in the area of agent-based modelling, a debate exists about the pros and cons of two perspectives, namely agent-based and population-based modelling. Agent-based models are often assumed

to produce more detailed, faithful behaviour, whereas population-based models abstract from such details to focus on global patterns (e.g., [2], [7], and [9]).

Given these similarities between the debate between macro- and microeconomics on the one hand, and the debate between population-based and agent-based modelling on the other hand, it makes sense to align the two debates. Hence, the goal of the current paper is to explore the differences and commonalities between population-based and agent-based modelling in an economical context. This will be done via a case study on the interplay between individual greed and the global economy.

This paper is structured as follows. In Section 2, the existing debate between agent-based and population-based modelling is briefly explained. In Section 3, both an agent-based and a population-based model are introduced for the example domain. In Section 4, a number of simulation results of both models are shown, and the similarities are discussed. Next, Section 5 provides a mathematical analysis on the models. Section 6 concludes the paper with a discussion.

2 Agent-Based versus Population-Based Modelling

The classical approaches to simulation of processes in which larger groups of agents are involved are population-based: a number of groups are distinguished (populations) and each of these populations is represented by a numerical variable indicating their number or density (within a given area) at a certain time point. The simulation model takes the form of a system of difference or differential equations expressing temporal relationships for the dynamics of these variables. Well-known classical examples of such population-based models address ecological processes, for example, predator-prey dynamics (e.g., [6], [12], [13] and [16]), and the dynamics of epidemics (e.g., [1], [6], and [8]). Such models can be studied by simulation and by using analysis techniques from mathematics and dynamical systems theory.

From the more recently developed agent system area it is often taken as a presupposition that simulations based on individual agents are a more natural or faithful way of modelling, and thus will provide better results (e.g., [2] and [7]). Although for larger numbers of agents such agent-based approaches are more expensive computationally than population-based approaches, such a presupposition may provide a justification of preferring their use over population-based approaches, in spite of the computational disadvantages. In other words, they are justified because the results are expected to deviate from the results of population-based simulation, and are considered more realistic. However, in contrast there is another silent assumption sometimes made, namely that for larger numbers of agents (in the limit), agent-based simulations approximate population-based simulations. This would indicate that for larger numbers of agents agent-based simulation just can be replaced by population-based simulation, which would weaken the justification for agent-based simulation discussed above. In, e.g., ([4; 9]), these considerations are explored for the domains of epidemics and crime displacement, respectively. The results put forward in these papers reveal several commonalities between both types of simulation, but also some differences. For example, for some specific parameter settings (concerning population size and rationality of the individual agents, among others), the results of population-based simulation seem to approximate those of agent-based simulation, whereas for other

situations some differences can be observed. Furthermore, as could be expected, the computation time of the populations-based simulations is shown to be much lower than that of the agent-based simulation.

In the next sections, similar issues are explored, but this time for a domain within economics. Comparative simulation experiments have been conducted based on different simulation models, both agent-based and population-based.

3 The Agent-Based and Population-Based Simulation Model

In this section, the two simulation models are introduced. First, an agent-based perspective is taken. The main idea behind this model is that the state of the global (world) economy influences the level of greed of the individual agents in the population, which is supposed to relate to the risk level of their investment decisions: in case the economic situation is positive, then people are tempted to take more risk. Moreover, the investment decisions of the individual agents in turn influence the global economy: in case agents become too greedy [10], this is assumed to have a negative impact on the economic situation, for example, due to higher numbers of bankruptcy. In addition, the state of the economy is assumed to be influenced by technological development which is driven by innovation. Inspired by these ideas, the interplay between agents' greed and the global economy is modelled as a dynamical system, in a way that has some similarity to predator-prey models in two variations: agent-based, where each agent has its own greed level, and population-based, where only an average greed level of the whole population is considered.

The agent-based model assumes n heterogeneous agents, which all interact within a certain economy. For each agent k , the individual greed is represented using a variable y_k , and the global economic situation is represented using a variable x . The complete set of variables and parameters used in the model is shown in Table 1.

Table 1. Variables and parameters used in the agent-based model

| | | |
|-------------------|---------------------------|-----------------------------------------------------------------------|
| Variables | x | World economy |
| | $y^{(1)}, \dots, y^{(n)}$ | Greed of individual agents |
| | z | Average greed of the agents (i.e., arithmetic mean of all $y^{(k)}$) |
| | TD | Technological development level |
| Parameters | a | Growth rate of the economy |
| | b | Decrease rate of the economy due to average greed |
| | c_1, \dots, c_n | Growth rate of an agent's greed based on the economy |
| | e_1, \dots, e_n | Decrease rate of an agent's greed |
| | inn | Innovation rate |

Based on these concepts, a system of difference equations was designed that consists of $n+3$ formulae; here (2) specifies a collection of n equations for each of the n agents, where each agent has its individual values for $y^{(k)}$, c_k and e_k :

(1) Updating the world economy

$$x_{new} = x_{old} + (a * x_{old} - b * x_{old} * z_{old}) * \Delta t$$

(2) Updating the greed of the agents

$$y_{new}^{(k)} = y_{old}^{(k)} + (c_k * b * x_{old} * y_{old}^{(k)} * (2 - y_{old}^{(k)}) / TD_{old} - e_k * y_{old}^{(k)}) * \Delta t \quad (\text{for all agents } k)$$

(3) Updating the technological development

$$TD_{new} = TD_{old} + inn * TD_{old} * \Delta t$$

(4) Aggregating greed

$$z_{old} = (\sum_k y^{(k)}_{old})/n$$

Table 2. Variables and parameters used in the population-based model

| | | |
|-------------------|------------|----------------------------------------------------------|
| Variables | <i>x</i> | World economy |
| | <i>y</i> | Average greed of the population |
| | <i>TD</i> | Technological development level |
| Parameters | <i>a</i> | Growth rate of the economy |
| | <i>b</i> | Decrease rate of the economy due to population greed |
| | <i>c</i> | Growth rate of the population greed based on the economy |
| | <i>e</i> | Decrease rate of the population greed |
| | <i>inn</i> | Innovation rate |

The population-based dynamical model is similar to the agent-based model, but the difference is that it abstracts from the differences of the individual agents. This is done by replacing the average greed *z* over all $y^{(k)}$ in formula (1) by one single variable *y* indicating the greed of the population as a whole, and using a single formula (2), which is only applied at the population level, in contrast to the collection of formulae (2) in the agent-based model, which are applied for all agents separately. The resulting population-based model is shown in Table 2 and in the formulae below.

(1) Updating world economy

$$x_{new} = x_{old} + (a * x_{old} - b * x_{old} * y_{old}) * \Delta t$$

(2) Updating the greed of the population

$$y_{new} = y_{old} + (c * b * x_{old} * y_{old} * (2 - y_{old}) / TD_{old} - e * y_{old}) * \Delta t$$

(3) Updating the technological development

$$TD_{new} = TD_{old} + inn * TD_{old} * \Delta t$$

Note that in differential equation format the agent-based and population-based dynamical model can be expressed by *n*+2, respectively 3 differential equations as shown in Table 3. Moreover, as the innovation rate *inn* is assumed constant over time, for both cases the differential equation for TD can be solved analytically with solution $TD(t) = TD(0) e^{inn * t}$.

Table 3. The two models expressed by *n*+2, respectively 3 differential equations

| Agent-based model | Population-based model |
|-----------------------------------------------------------------------|----------------------------------|
| $dx/dt = ax - bxz$ | $dx/dt = ax - bxy$ |
| $d y^{(k)} / dt = (c_k b x y^{(k)} (2 - y^{(k)}) / TD) - e_k y^{(k)}$ | $dy/dt = (cb xy(2-y) / TD) - ey$ |
| $dTD/dt = inn TD$ | $dTD/dt = inn TD$ |
| $z = (\sum_k y^{(k)})/n$ | |

4 Simulation Results

Based on the model introduced above, a number of simulation experiments have been performed under different parameter settings (with population size varying from 2 to 400 agents), both for the agent-based and for the population-based case. Below, a number of them are described. First an agent-based simulation experiment is described. In this first experiment, 25 agents were involved. The initial settings used for the variables and parameters involved in the experiment are shown in Table 4.

Table 4. Initial settings for variables and parameters

| Parameter | Value | Variable | Initial value |
|------------|----------------------------|------------|----------------------|
| <i>a</i> | 1.5 | <i>x</i> | 5 |
| <i>b</i> | 5.8 | <i>y</i> | random in [0.2, 0.3] |
| <i>c</i> | random in [0.0260, 0.0274] | <i>TD</i> | 1 |
| <i>e</i> | random in [0.85, 0.89] | | |
| <i>inn</i> | 0.01 | Δt | 0.1 |

The results of the simulations are shown in Figure 1a and 1b. In Figure 1a, time is on the horizontal axis and the value of the world economy is represented on the vertical axis. It is evident from the graph that the economy grows as time increases (but fluctuating continuously). Figure 1b shows the individual greed values of all 25 agents. As can be seen they fluctuate within a bandwidth of about 25% with lowest points between about 0.1 and 0.15, and highest points around 0.45. The pattern of the average greed over all 25 agents is shown in Figure 1c.

For the population-based simulation, all the parameter settings are the same as in Table 4, except parameters *y*, *c* and *e*. The values for parameters *y*, *c* and *e* used in the population-based simulation were determined on the basis of the settings for the agent-based simulations by taking the average *y*, *c* and *e* for all fifty agents:

$$y = (\sum_k y_k)/n \qquad c = (\sum_k c_k)/n \qquad e = (\sum_k e_k)/n$$

The results of the population-based simulations are shown in Figure 2a (economy) and 2b (greed). As can be seen from these figures, the results approximate the results for the agent-based simulation. The difference of the world economy for the population-based and agent-based simulation (averaged over all time points) turns out to be 0.112, and the difference between the average greed of the 25 agents in the agent-based simulation and the greed for the population-based simulation is 0.005.

In addition, a number of simulation runs have been performed for other population sizes. Figure 3a displays the (maximum and average) difference between the world economy in the agent-based model and the world economy in the population-based model for various population sizes. Similarly, Figure 3b displays the difference between the average greed in the agent-based model and the greed in the population-based model for various population sizes. The red line indicates the maximum value and the blue line the average value over all time points. As the figures indicate, all differences approximate a value that is close to 0 as the population size increases. Although the results of these particular simulation experiments should not be over-generalised, this is a first indication that for higher numbers of agents, the results of the agent-based model can be approximated by those of the population-based model.

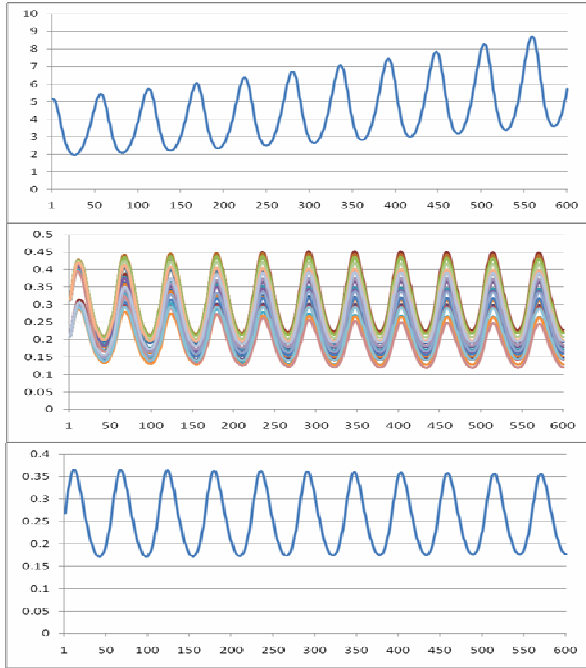


Fig. 1. Agent-based simulation results:
a) world economy, b) individual greed of 25 agents, and c) average greed (over 25 agents)

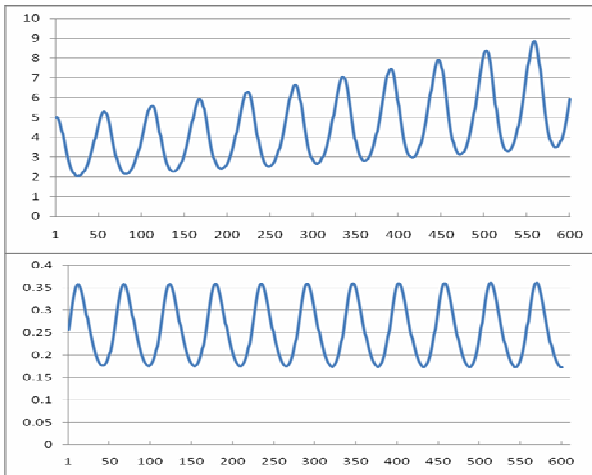


Fig. 2. Population-based simulation results: a) world economy, and b) greed

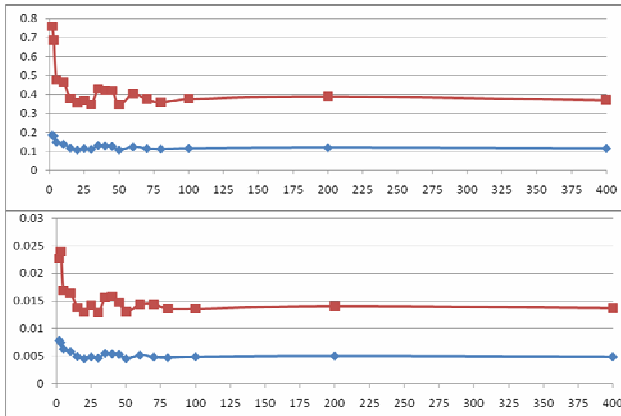


Fig. 3. Difference between both models for various population sizes:
a) world economy, and b) greed

5 Mathematical Analysis

In this section a mathematical analysis is presented concerning the conditions under which partial or full equilibria occur; it is assumed that the parameters a , b , c and e are nonzero. For an overview of the equilibria results, see Table 5.

Dynamics of the economy. The economy grows when $dx/dt > 0$ and shrinks when $dx/dt < 0$; it is in equilibrium when $dx/dt = 0$. Assuming x nonzero, according to equation (1) for the population-based model, this can be related to the value of the greed as follows

| | |
|-------------------------------|---------------------------------------------------------------------------------------------|
| economy grows | $dx/dt > 0 \Leftrightarrow ax - bxy > 0 \Leftrightarrow a - by > 0 \Leftrightarrow y < a/b$ |
| economy shrinks | $dx/dt < 0 \Leftrightarrow ax - bxy < 0 \Leftrightarrow a - by < 0 \Leftrightarrow y > a/b$ |
| economy in equilibrium | $dx/dt = 0 \Leftrightarrow ax - bxy = 0 \Leftrightarrow a - by = 0 \Leftrightarrow y = a/b$ |

So, as soon as the greed exceeds a/b the economy will shrink (for example, due to too many bankruptcies), until the greed has gone below this value. This indeed can be observed in the simulation traces. For the agent-based model similar criteria can be derived, but then relating to the average greed z instead of y .

Full Equilibria for the Population-Based Model. The first issue to be analysed is whether (nonzero) equilibria exist for the whole population-based model, and if so, under which conditions. This can be analysed by considering that x , y and TD are constant and nonzero. For x constant above it was derived from (1) that the criterion is $y = a/b$. For TD constant the criterion is $inn = 0$ as immediately follows from (3). The criterion for $dy/dt = 0$ can be derived from (2) as follows

$$dy/dt = (cbxy(2-y)/TD - ey) = 0 \Rightarrow cbx(2-y)/TD = e \Rightarrow x = (e / ((2b-a)c)) TD$$

This provides the conditions for a full equilibrium

$$(1) \ y = a/b \qquad (2) \ x = (e / ((2b-a)c)) TD \qquad (3) \ inn = 0$$

It turns out that for any nonzero setting for the parameters a , b , c and e and for setting $inn = 0$ for the innovation parameter and for any value of TD a nontrivial equilibrium is (only) possible with values as indicated above. Note that this shows that for inn nonzero a nontrivial full equilibrium is not possible, as TD will change over time. However, partial equilibria for greed still may be possible. This will be analysed next

Equilibria for greed in the population-based model. Suppose that the innovation inn is nonzero. In this case it cannot be expected that technological development TD and economy x stay at constant nonzero values. However still for the greed variable y an equilibrium may exist. From the second equation (2) by putting $dy/dt = 0$ it follows

$$cbx(2-y)/TD = e \Rightarrow x = \alpha TD \quad \text{with } \alpha = e/cb(2-y)$$

By filling this in differential equation (1) it follows

$$d \alpha TD /dt = a \alpha TD - b \alpha TD y \Rightarrow d TD /dt = (a - by) TD$$

By differential equation (3) it can be derived

$$d TD /dt = (a - by) TD = inn TD \Rightarrow (a - by) = inn \Rightarrow y = (a - inn)/b$$

Note that for $inn = 0$ this also includes the result for the full equilibrium obtained earlier. Moreover, as the equation for TD can be solved analytically, and $x = \alpha TD$, also an explicit solution for x can be obtained:

$$TD(t) = TD(0) e^{inn t} \quad x(t) = \alpha TD(t) = \alpha TD(0) e^{inn t} = x(0) e^{inn t}$$

Here α can be expressed in the parameters as follows:

$$\alpha = e/cb(2-y) = e/cb(2-(a-inn)/b) = (e/c)/(2b-a+inn)$$

This shows that according to the model greed can be in an equilibrium $y = (a - inn)/b$, in which case the economy shows a monotonic exponential growth.

Full Equilibria for the agent-based model. Similar to the approach followed above:

- (1) $dx/dt = (ax - bxz) = 0$
- (2) $d y^{(k)} /dt = (c_k b x y^{(k)} (2 - y^{(k)}) / TD - e_k y^{(k)}) = 0$ (for all agents k)
- (3) $dTD/dt = inn TD = 0$
- (4) $z = (\sum_k y^{(k)})/n$

A full equilibrium can be expressed by the following equilibria equations:

- (1) $ax = bxz$
- (2) $c_k b x y^{(k)} (2 - y^{(k)}) / TD = e_k y^{(k)}$
- (3) $inn TD = 0$
- (4) $z = (\sum_k y^{(k)})/n$

It is assumed that a , b , c_k and e_k are nonzero. One trivial solution is $x = y^{(k)} = 0$. Assuming that x , $y^{(k)}$ and TD all are nonzero, the equations (1) to (3) are simplified:

- (1) $a = bz$
- (2) $c_k b x (2 - y^{(k)}) / TD = e_k$
- (3) $inn = 0$
- (4) $z = (\sum_k y^{(k)})/n$

This provides

- (1) $z = a/b$
- (2) $y^{(k)} = 2 - e_k TD / (c_k b x)$
- (3) $inn = 0$
- (4) $z = (\sum_k y^{(k)})/n$

From the second, first and last equation it follows that

$$a/b = (\sum_k y^{(k)})/n = (\sum_k (2 - e_k TD/(c_k bx)))/n = 2 - \sum_k (e_k TD/(c_k bx))/n = 2 - (TD/bx) (\sum_k (e_k/c_k))/n \Rightarrow x = TD \sum_k (e_k/c_k)/(2b - a)n$$

From this the values for the $y^{(j)}$ can be determined:

$$\begin{aligned} y^{(j)} &= 2 - e_j TD/(c_j bx) = 2 - e_j TD/(c_j b TD \sum_k (e_k/c_k)/(2b - a)n) \\ &= 2 - e_j/(c_j b \sum_k (e_k/c_k)/(2b - a)n) = 2 - e_j(2b - a)n/(c_j b \sum_k (e_k/c_k)) \\ &= 2 - e_j(2 - (a/b))n/(c_j \sum_k (e_k/c_k)) = 2 - (2 - (a/b))n/(\sum_k (e_k/e_j)(c_j/c_k)) \end{aligned}$$

It turns out that for any nonzero setting for the parameters a, b, c_k and e_k and for setting $inn = 0$ for the innovation parameter, and for any value of TD a nontrivial equilibrium is (only) possible with values as indicated above.

Equilibria for greed for the agent-based model. From the second equation $c_k bx (2 - y^{(k)})/TD = e_k$ with $y^{(k)}$ constant it follows that $x = \alpha_k TD$ with α_k the constant $\alpha_k = e_k/c_k b (2 - y^{(k)})$ which apparently does not depend on k , as both x and TD do not depend on k , so the subscript in α_k can be left out. Filling this in (1) provides:

$$d \alpha TD/dt = (a \alpha TD - b \alpha TD z) \Rightarrow d TD/dt = (a - bz) TD$$

By differential equation (3) it can be derived

$$dTD/dt = (a - bz) TD = inn TD \Rightarrow (a - bz) = inn \Rightarrow z = (a - inn)/b$$

Now the equilibrium values for $y^{(j)}$ can be determined as follows.

$$\alpha = e_k/c_k b (2 - y^{(k)}) \Rightarrow 2 - y^{(k)} = e_k/\alpha c_k b \Rightarrow y^{(k)} = 2 - e_k/c_k \alpha b$$

Next the value of α is determined $z = (\sum_k y^{(k)})/n = \sum_k (2 - e_k/c_k \alpha b)/n = 2 - (1/\alpha bn) \sum_k e_k/c_k$. Since $z = (a - inn)/b$ it follows

$$\begin{aligned} (a - inn)/b &= 2 - (1/\alpha bn) \sum_k e_k/c_k \Rightarrow (1/n \alpha) \sum_k e_k/c_k = 2b - (a - inn) \Rightarrow \\ \sum_k e_k/c_k &= (2b - (a - inn)) n \alpha \Rightarrow \alpha = \sum_k (e_k/c_k)/(2b - (a - inn))n \end{aligned}$$

Given this value for α the equilibrium values for the greed $y^{(j)}$ are

$$\begin{aligned} y^{(j)} &= 2 - e_j/c_j \alpha b = 2 - e_j/b c_j \sum_k (e_k/c_k)/(2b - (a - inn))n \\ &= 2 - (2 - (a - inn)/b) n / \sum_k (e_k c_j/e_j c_k) \end{aligned}$$

Table 5. Overview of the equilibria of the two models

| | Agent-based model | Population-based model |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| Full equilibrium | $inn = 0$ $x = (1/(2b - a)) (\sum_k (e_k/c_k)/n) TD$ $z = a/b$ $y^{(j)} = 2 - (2 - (a/b)) n / \sum_k (e_k/e_j)(c_j/c_k)$ | $inn = 0$ $x = (1/(2b - a))(e/c) TD$ $y = a/b$ |
| Partial equilibrium for greed | $TD(t) = TD(0) e^{inn t}$ $x(t) = (1/(2b - a + inn)) (\sum_k (e_k/c_k)/n) TD(0) e^{inn t}$ $z = (a - inn)/b$ $y^{(j)} = 2 - ((a - inn)/b) n / \sum_k (e_k/e_j)(c_j/c_k)$ | $TD(t) = TD(0) e^{inn t}$ $x(t) = (1/(2b - a + inn)) (e/c) TD(0) e^{inn t}$ $y = (a - inn)/b$ |

6 Discussion

This paper discusses similarities and dissimilarities between agent-based models and population-based models in behavioural economics. Inspired by variants of predator-prey models (e.g., [6], [12], [13], and [16]), a dynamic behavioral economical model was developed for the relationship between individual agents' greed and the global economy. Simulation experiments for different population sizes were performed for both an agent-based and a population-based model. For both cases the results show that the world economy grows in a fluctuating manner over time and the average greed of the agents fluctuates between 0.1 and 0.45. A mathematical analysis was performed for both, showing the conditions under which equilibria occur.

It turned out that, in particular for large population sizes, the differences in the economy and average greed between agent-based and population based simulations are close to zero. In different domains, in [4] and [9], under certain conditions similar results were obtained. In literature on agent-based simulation such as in (e.g., [2] and [7]), it is argued that although agent-based modelling approaches are more expensive computationally than population-based modelling approaches, they are preferable due to more accuracy. In contrast to this, the results in the current paper indicate that for the considered domain the agent-based approaches can be closely approximated by population-based simulations. On the other hand, for cases with a rather small number n of agents the population-based approach may be inadequate. This may raise the question whether a more differentiated point of view in the debate can be considered, namely that for numbers of n agents exceeding a certain N , population-based models are as adequate as agent-based models, whereas for $n < N$ agent-based models are more adequate. A challenge may be to determine this number N for different cases.

For future work, more differentiated personality aspects will be included in the agent model, concerning risk profile and emotions (e.g., feeling insecure) involved, depending upon which decisions are made for the investment (in banking products or stock market). A further aim is to develop a web-based business application incorporating a virtual agent that will interact with a client and regulate the emotions.

References

1. Anderson, R.A., May, R.M.: Infectious Diseases of Humans: Dynamics and Control. Oxford University Press, Oxford (1992)
2. Antunes, L., Paolucci, M., Norling, E. (eds.): MABS 2007. LNCS (LNAI), vol. 5003. Springer, Heidelberg (2008)
3. Blanchard, O., Fischer, S.: Lectures in Macroeconomics. MIT Press, Cambridge (1989)
4. Bosse, T., Gerritsen, C., Hoogendoorn, M., Jaffry, S.W., Treur, J.: Comparison of Agent-Based and Population-Based Simulations of Displacement of Crime. In: Jain, L., et al. (eds.) Proceedings of the 8th IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2008, pp. 469–476. IEEE Computer Society Press, Los Alamitos (2008)
5. Brandstätter, H.: Should Economic Psychology Care about Personality Structure? Journal of Economic Psychology 14, 473–494 (1993)
6. Burghes, D.N., Borrie, M.S.: Modelling with Differential Equations. John Wiley, Chichester (1981)

7. David, N., Sichman, J.S. (eds.): *Multi-Agent-Based Simulation IX*. LNCS, vol. 5269. Springer, Heidelberg (2009)
8. Ellner, S.P., Guckenheimer, J.: *Dynamic Models in Biology*. Princeton University Press, Princeton (2006)
9. Jaffry, S.W., Treur, J.: Agent-Based and Population-Based Simulation: A Comparative Case Study for Epidemics. In: Louca, L.S., Chrysanthou, Y., Oplatkova, Z., Al-Begain, K. (eds.) *Proc. of the 22nd European Conference on Modelling and Simulation, ECMS 2008*. European Council on Modeling and Simulation, pp. 123–130 (2008)
10. Kasser, T., Cohn, S., Kanner, A., Ryan, R.: Some costs of American corporate capitalism: A psychological exploration of value and goal conflicts. *Psychological Inquiry* 18, 1–22 (2007)
11. Kreps, D.M.: *A Course in Microeconomic Theory*. Princeton University Press, Princeton (1990)
12. Lotka, A.J.: *Elements of Physical Biology*. Reprinted by Dover in 1956 as *Elements of Mathematical Biology* (1924)
13. Maynard, S.: *Models in Ecology*. Cambridge University Press, Cambridge (1974)
14. Simon, H.A.: Behavioural Economics. In: *The New Palgrave: A Dictionary of Economics*. MacMillan, London (1987)
15. Tesfatsion, L.: Agent-based computational economics: Growing economies from the bottom up. *Artificial Life* 8, 55–82 (2002)
16. Volterra, V.: Fluctuations in the abundance of a species considered mathematically. *Nature* 118, 558–560 (1926)

Modeling and Verifying Agent-Based Communities of Web Services

Wei Wan¹, Jamal Bentahar², and Abdessamad Ben Hamza²

¹ Department of Electrical and Computer Engineering, Concordia University

² Concordia Institute for Information Systems Engineering, Concordia University
w_wan@encs.concordia.ca, {bentahar,hamza}@ciise.concordia.ca

Abstract. Communities of web services are virtual spaces that can dynamically gather different web services having complementary functionalities in order to provide composite services. In the last two years, some approaches have been proposed using multi-agent systems to organize communities of web services. This trend has increased the flexibility but also the system complexity. The system becomes hard to check by simply inspecting its model. Therefore, model checking, which is a well-established formal technique for verifying communication and cooperation in multi-agent systems, is used in this paper to verify the system correctness in terms of satisfying desirable properties. The approach presented in the paper is used to verify communities of web services modeled in UML activity diagram. We first translate the activity diagram into an interpreted system model using predefined transformation rules. Specifications are expressed as formulae in a logic extending the Computation Tree Logic *CTL** with agent commitments needed for their communication. Then, both the model and formulae are used as inputs for the multi-agent symbolic model checker MCMAS. We illustrate our approach with a short case study, in which we show how communication properties of simulated communities are verified.

Keywords: Multi-Agent Systems, Communities of Web Services, UML, Model Checking, MCMAS.

1 Introduction

As the internet becomes more and more prevalent, communities of web services, which are large scale virtual networks of web services, attract more and more attention [1,2]. In the last two years, these communities started to move toward "agent-like" models that include largely independent agent-based web services. However, this merging results in more complex systems where functional and non-functional properties cannot be easily checked by simply inspecting the system model. Since it is very expensive to modify communities of web services that have been deployed, it is desirable to have methods available for the verification of communities' properties earlier in the design phases. Model checking [5] is a suitable solution in this case because it is a formal technique allowing the automatic verification of the systems design against specific properties that capture the requirements.

In nineties, Researchers have put forward in [7] an application of model checking within the context of the logic of knowledge. After that, several approaches have been proposed for model checking multi-agent systems. In [16], Wooldridge et al. have proposed an imperative programming language, MABLE to specify multi-agent systems along with a Belief-Desire-Intention (*BDI*) logic to express the properties. SPIN, an automata-based model checker has been used to verify if the specified MABLE model satisfies the expressed properties. Another method based on the SPIN model checker has been developed in [3] using AgentSpeak(F) Language, a *BDI* logic-based programming language [12]. As a model grows, automata-based model checking can face a serious state explosion problem. One technique to avoid this problem is symbolic model checking based on Ordered Binary Decision Diagrams (OBDDs). NuSMV [4], MCK [14] and MCMAS [9] are examples of model checkers using this approach. NuSMV supports both Linear Temporal Logic (*LTL*) and branching time logic (*CTL*). MCK works on a particular input model of synchronous interpreted systems of knowledge. The specification formulae in MCK can be either *LTL* or *CTL* augmented with knowledge. Similar to MCK, in MCMAS, models are described into a modular language called Interpreted Systems Programming Language (ISPL). Although the framework of interpreted systems is powerful and popular in multi-agent systems, it cannot be directly used by designers to describe business and industrial systems. For this type of applications, it deems appropriate to use suitable modeling languages such as UML (Unified Modeling Language).

The motivation of this paper is to build the connection among UML, agent-based communities, and symbolic model checking so that we can use existing model checkers, like MCMAS, to check these communities' models directly. We propose an approach based upon symbolic model checking to verify communities presented by UML activity diagram, which shows the activities and flow of control for the model [15]. We formalize agent-based communities of web services with the execution semantics of UML activity diagram. We adopt CTL^{*CA} proposed in [2] for communicating agents as the logic for specifying the properties to be checked. We use the MCMAS model checker in our symbolic approach for the verification of communities of web services. There are two reasons behind choosing MCMAS: 1) unlike NuSMV and MCK, MCMAS supports directly agent specifications we need for agent-based communities of web services; and 2) in terms of the adopted specification language, MCMAS is the closest to CTL^{*CA} . We experiment this approach with an implementation to verify the *PNAWS* protocol (Persuasion/Negotiation protocol for Agent-based Web Services) [2]. *PNAWS* is a communication protocol used by agent-based web services to negotiate joining a given community.

The structure of this paper is as follows: In Section 2, we present an overview of our model checking approach and explain how we formalize the activity diagram to represent communicating agent-based web services. The specification language CTL^{*CA} logic for communicating agents will be also discussed. In Section 3, we define the rules for mapping and transforming formalized activity diagram model and properties specification into the Interpreted Systems Programming Language

(ISPL), which is used as the input language of the MCMAS model checker. Section 4 presents the experimental results of verifying the *PNAWS* protocol with MCMAS. Finally, we summarize our work and discuss future work in Section 5.

2 Modeling/Specifying Communities of Web Services

2.1 Approach Overview

Model checking is a three-step process [5]: modeling, specification, and verification. We use UML activity diagrams to model the system, CTL^{*CA} as specification language to state the properties that the design must satisfy, and symbolic model checking with OBDDs for verification.

Fig. 1 illustrates our general approach. It starts with modeling communities of web services as activity diagrams and specifying the properties as formal requirements. The modeled system and formalized specifications are read as inputs by our automatic transformation engine, which uses transformation definitions (rules) to map the input model and specifications (properties) into the ISPL model and formulae. Finally, the MCMAS model checker verifies the ISPL model against the formulae. Witnesses are generated if the formulae are true (i.e. the properties are satisfied); otherwise, counterexamples are generated.

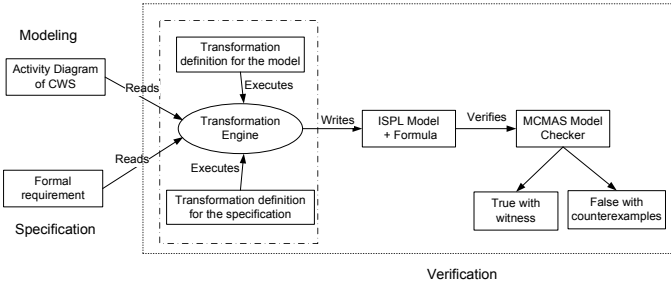


Fig. 1. Structure of model checking communities of web services (CWSs)

Before using this approach, an issue should be resolved: the activity diagram modeling the community of web services can have an infinite state space, while symbolic model checking requires the state space to be finite. To convert activity diagram from infinite to finite state space, Eshuis and Wieringa have proposed in [6] some techniques to remove unbounded states, which do not have a maximum number of their *active instances*. We adopt this method in our approach.

2.2 UML Activity Diagrams

A UML activity diagram shows the activities and flow of control for the model [15]. Activity states are represented with rounded rectangles, a black solid circle stands for an initial node and a black in-out circle is a final node. A diamond

represents a decision or merging state. A bar shows an activity that splits a flow into several concurrent flows or an activity that synchronizes several concurrent flows and joins them into one single flow.

Fig 2 shows an activity diagram for a concrete example of *PNAWS* protocol model [2]. According to this protocol, agent-based web services interact with each other in a negotiation setting. Our example diagram presents the Master web service (MWS) agent’s behavior of inviting a Slave web service to join its community and the Slave web service (SWS) agent’s behavior of negotiating the joining contract. The MWS, which represents the community, starts the session by sending the invitation. The SWS can either accept or refuse. If the SWS accepts, the session will end with successful invitation. Otherwise, the MWS will defend the invitation proposal with negotiation arguments. Then, if the defence is not definitely accepted or refused, the MWS and SWS will start a negotiation process consisting of a sequence of challenge/justification and attack until they achieve an agreement, refusal or timeout.

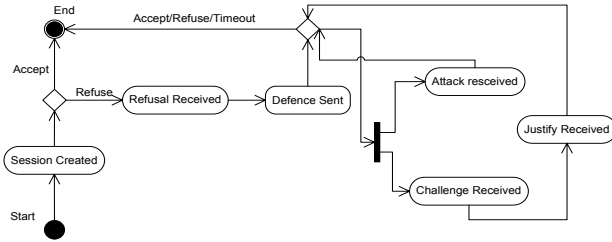


Fig. 2. Activity diagram of *PNAWS* protocol

The activity diagram used in this paper describes the behavior of agents that interact with each other. These agents perform certain actions according to the protocol they use, which is a set of rules describing the allowed communicative acts in different situations. An agent has beliefs, goals, and intentions that are stored in a database accessible to the agent but external to the activity diagram. Activity states represent activities performed by certain agents, such as accepting or refusing a proposal. A transition from one state to another is triggered by a set of internal (by agent’s own actions) or external (by other agents’ actions) activities. We use \xrightarrow{Act} to represent the transition relation.

In order to use activity diagrams in our symbolic model checking approach, we need to define their formal semantics. Because we associated them to agent communication protocols, a suitable solution would be to define a formal *agent hypergraph* from the notion of *activity hypergraph* used to model check activity diagrams [6]. The idea behind an *agent hypergraph* is to capture the execution structure of the communication protocol among agent-based web services. We use CTL^{*CA} model to represent the communicative acts agents use when communicating. These communicative acts are defined as action performed on public commitments the agents make. For example, by inviting a Slave web service to join a community, the Master *creates* a new public commitment and accepting

this invitation means *accepting* the content of this public commitment. An *agent hypergraph* is defined as a tuple: $\langle S, s_0, Ag, Act, \xrightarrow{Act}, V_{PC} \rangle$, where:

- S is a set of all possible states in the system. There are three kinds of states in this set: one initial state, at least one final state, and none or several activity states which are not initial state or final states.
- s_0 is the initial state.
- Ag is a non-empty set of agents.
- Act is a set of allowed actions agents can perform.
- $\xrightarrow{Act} \subseteq S \times Ag \times S$ is the transition relation. We write $s_i, Ag_n \xrightarrow{Act_k} s_j$ to express how the agent Ag_n evolves from one state s_i to another state s_j by performing the action Act_k .
- $V_{PC} : S \rightarrow 2^C$ is a function associating to each state the set of public commitments made in this state, where C is the set of all public commitments.

2.3 Logic for Specification - CTL^{*CA}

Syntax. We use CTL^{*CA} [2] to specify the properties our agent-based communities of web services should satisfy. CTL^{*CA} extends CTL^* [5] by adding public commitments and action formulae. This logic supports two kinds of formulae: state formulae \mathcal{S} evaluated over states and path formulae \mathcal{P} evaluated over paths that are infinite sequences of states. We use p, p_1, p_2, \dots to range over the set of atomic propositions Φ_p and ϕ_1, ϕ_2, \dots to range over path formulae. The syntax of this logic is given in Table 1.

Table 1. The Syntax of CTL^{*CA} Logic

| |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\mathcal{S} ::= p \neg \mathcal{S} \mathcal{S} \vee \mathcal{S} A\mathcal{P} E\mathcal{P} PC(Ag_1, Ag_2, t, \mathcal{P})$ |
| $\mathcal{P} ::= \mathcal{S} \mathcal{P} \vee \mathcal{P} X^+ \mathcal{P} X^- \mathcal{P} \mathcal{P}U^+ \mathcal{P} \mathcal{P}U^- \mathcal{P} \mathcal{P} \therefore \mathcal{P}$ $ Act_k(Ag_n, PC(Ag_1, Ag_2, t, \mathcal{P}))$ |

The temporal operator $X^+ \phi_1$ means in the next state ϕ_1 is true, $X^- \phi_1$ means in the previous state ϕ_1 is true, $\phi_1 U^+ \phi_2$ means ϕ_1 is true until ϕ_2 becomes true and $\phi_1 U^- \phi_2$ means ϕ_1 is true until ϕ_2 was true. A stands for the universal path quantifier and E stands for the existential path quantifier. The formula $\phi_1 \therefore \phi_2$ means that ϕ_1 is an argument for ϕ_2 and is read as: ϕ_1 so ϕ_2 . This operator introduces argumentation as a logical relation between path formulae.

The formula $PC(Ag_1, Ag_2, t, \phi_1)$ is the public commitment made by agent Ag_1 at the moment t towards agent Ag_2 saying that the path formula ϕ_1 is true. $Act_k(Ag_n, PC(Ag_1, Ag_2, t, \phi_1))$ means that agent Ag_n ($n \in \{1, 2\}$) performs an action Act_k on the commitment made by Ag_1 towards Ag_2 . The set of actions performed on commitments may change to suit different systems. For communities of web services, we use *Create*, *Accept*, *Refuse*, *Defend*, *Challenge*, *Justify*, and *Attack*.

Formal Semantics. The formal model M associated to this logic corresponds exactly to our *agent hypergraph* defined above. Because of space limit and to

focus more on the verification issue, which is the main contribution of this paper, here we only specify the semantics of the argument and commitment operators. The semantics of CTL^{*CA} state formulae is as usual (semantics of CTL^*). A path satisfies a state formula if the initial state in the path does so. Along a path x^i , which starts at state s_i , $\phi_1 \therefore \phi_2$ holds iff ϕ_1 is true and in the next state through the same path if ϕ_1 holds then ϕ_2 holds too. Formally (\Rightarrow stands for material implication):

$$x^i \models_M \phi_1 \therefore \phi_2 \text{ iff } x^i \models_M \phi_1 \text{ and } x^{i+1} \models_M \phi_1 \Rightarrow \phi_2$$

A state s_i satisfies $PC(Ag_1, Ag_2, t, \phi_1)$ iff the commitment is in this state and there is a path along which the commitment content holds. Formally:

$$s_i \models_M PC(Ag_1, Ag_2, t, \phi_1) \text{ iff } PC(Ag_1, Ag_2, t, \phi_1) \in V_{PC}(s_i) \text{ and } s_i \models_M E\phi_1$$

A path x^i satisfies $Act_k(Ag_n, PC(Ag_1, Ag_2, t, \phi_1))$ iff Act_k is in the label of the first transition on this path and in the past¹ $PC(Ag_1, Ag_2, t, \phi_1)$ holds along the same path. Formally:

$$x^i \models_M Act_k(Ag_n, PC(Ag_1, Ag_2, t, \phi_1)) \text{ iff } s_i, Ag_n \xrightarrow{Act_k} s_{i+1} \text{ and } x^i \models_M F^- PC(Ag_1, Ag_2, t, \phi_1)$$

3 Verification

In this section, we will use the *PNAWS* protocol presented in Section 2.2 to show how our verification approach works. As discussed earlier, we use the MCMAS model checker. In MCMAS, multi-agent systems are described by the Interpreted Systems Programming Language (ISPL), where the system is distinguished into two types of agents: environment agent, which is used to describe boundary conditions and infrastructures, and standard agents. ISPL can also be used to define atomic propositions, action formulae and the specification of properties to be checked. To automatically use this model checker to verify the communication protocol of community of web services, we define a mapping between our *agent hypergraph* and ISPL and encode our CTL^{*CA} formulae in MCMAS.

3.1 Mapping and Transforming Agent Hypergraph to ISPL

In MCMAS, Each agent is composed by: a set of local states, a set of actions, a rule (protocol) describing which action can be performed by an agent, and evolution functions that describe how the local states of the agents evolve based on their current local states and agents' actions [9]. The mapping from our *agent hypergraph* to ISPL is defined by the following rules:

1. *S – to – LocalState*: Every state in the *agent hypergraph* is mapped to the ISPL environment agent, a local state with the same name.
2. *Ag – to – Agent*: Every agent in the *agent hypergraph* is mapped to an ISPL agent with the same name. A special agent environment should also be added to the ISPL agent list.

¹ The past operator F^- is an abbreviation and defined as follows: $F^- \phi_1 \equiv trueU^- \phi_1$.

3. $V_{PC} - to - LocalValue$: V_{PC} is transformed to local values of the agent that creates the public commitments. The values can be bounded integers, Booleans or enumerations based on the types of these commitments.
4. $Act - to - action/rule$: Every action in the *agent hypergraph* is converted to an ISPL action list of an agent that can execute the action.
5. $\xrightarrow{Act} - to - evolution$: \xrightarrow{Act} is translated into an ISPL agent evolution list. For example if we have $s_i, Ag_n \xrightarrow{Act_k} s_j$, the Ag_n 's evolution in ISPL will be:
`state = s(j) if state = s(i) and Action = Act(k);`
6. $s_0 - to - Init$: s_0 is mapped to an ISPL initial state.

Based on these mapping rules, we use *PNAWS* as an example to transform the associated *agent hypergraph* into ISPL. In communities of web services, the system includes two types of agents: Master agent and Slave agent. We add an Environment agent to describe the system boundary conditions and infrastructures. Environment agent is a special agent in ISPL system that provides observable variables that can be accessed by other agents. Every agent starts with declaration of local variables. The first mapping rule is used to define the local variables of agent. We declare a state variable to list all possible states in the system:

```
Vars: State: {WaitingforCreate, RefuseReceive...}; end Vars
```

Actions an agent can perform are constructed into *Actions* section of ISPL file and follow the 4th mapping rule. We also add “null” action to stand for no action. In ISPL *Protocol* section, we give the permitted actions in each state. Transitions are defined in ISPL *Evolution* section to show the states change based on the 5th mapping rule.

```
Protocol:
```

```
state = WaitingforCreate: {Create}; ...
```

```
end Protocol
```

```
Evolution:
```

```
state = ChallengeReceived if state = DefenceSend  
and Slave1.Action=Challenge; ...
```

```
end evolution
```

Moreover, we declare a set of initial states. The system starts at state waiting for creating a protocol session with all the counter register reset.

```
InitStates
```

```
(Environment.state = WaitingforCreated) and
```

```
(Master.commitments = toCreate) and
```

```
(Environment.attackCount = 0) and
```

```
(Environment.challengeCount = 0);
```

```
end InitStates
```

3.2 Encoding Specifications in ISPL

ISPL specifies both the model and properties. It supports CTL and ATL. Our specifications are expressed by CTL^{*CA} , which extends CTL^* . Therefore, the

basic operators are similar to CTL and we can directly use them in ISPL. For the new operators in CTL^{*CA} , we define rules to convert them into ISPL.

To translate $\phi_1 \dot{\cdot} \phi_2$ formula into ISPL, we need to declare two variables over the system: **a1** and **a2** to stand for ϕ_1 and ϕ_2 . We also need to define the equivalent formula according to the semantics given in Section 2.3, where **X** stands for next and \rightarrow stands for implication.

```

Evaluation
  a1; a2; ...
end Evaluation
Formula a1 and X(a1 -> a2); end Formula
    
```

For the formulae $PC(Ag_1, Ag_2, t, \phi_1)$ and $Act_k(Ag_n, PC(Ag_1, Ag_2, t, \phi_1))$, the semantics is already encoded in ISPL as V_{PC} and \xrightarrow{Act} are already translated by the 3rd and 5th rules. We just need to define a local value **a1** in agent **Ag1**'s definition to present the commitment, then create the action Act_k over this commitment. The moment **t** is declared in Environment agent because both agents **Ag1** and **Ag2** need to access it at that moment. The code below is an example of the **Create** action (i.e. sending an invitation).

```

Evolution:
  Lvar = Ag2Lvar if moment = t and Ag1.Action = Create and a1;
  ...
end evolution
    
```

In order to verify the model, we first define some atomic propositions over the system. Thereby, the propositional formulae, which we need to check by MCMAS, are defined based on these propositions.

4 Experimental Results

We have implemented the mapping rules and the *PNAWS* case study scenario along with the specifications in ISPL and verified them with MCMAS. We formalize various properties of compliance for *PNAWS* protocol. Here are some examples, where **G** means globally, **F** means in the future and **A** and **E** are the universal and existential quantifiers.

1. Termination: *PNAWS* always terminates.

AG termination

Termination is an atomic proposition for termination state of the protocol. Intuitively, this property should hold because with finite states and restrained unbound states, a protocol will end.

2. Soundness: The protocol is correct. An example of soundness is: if there is a challenge, a justification will follow in the future.

AG (challenge -> EF justify)

3. Reachability: certain states are reachable through any possible sequence of transitions, starting from the initial state. For example, if there is a refusal for an invitation, the protocol will reach defense state.

AG (refusal -> EF defense)

4. Liveness: Liveness means something good will eventually happen. An example of liveness is: if there is a negotiation, an acceptance will eventually follow in the future.

AG (Attack or Challenge \rightarrow EF Accept)

Our system was running on Windows Vista Home Premium on Inter Core 2 Duo CPU T6400 2.00GHz with 3.0GB memory. We used different numbers of slave agents in the systems to monitor the changes as the models grow. Experimental results are presented in Table 2. The first column indicates the numbers of Slave Agents in the system. The number of actual reachable states in the corresponding model is shown in column two. The third column reports the total number of nodes that were requested and obtained during reordering and verification process. Memory usage and approximate execution time are listed in column four and column five. The rest of columns show that the properties are satisfied. The results clearly show that the state space grows exponentially, but thanks to symbolic model checking the execution time is low.

Table 2. Experimental Results

| n Slave Agents | Reachable states | Nodes allocated | Memory in use (MB) | Execution time (\approx sec) | Property | | | |
|------------------|------------------|-----------------|--------------------|---------------------------------|----------|------|------|------|
| | | | | | 1 | 2 | 3 | 4 |
| 1 | 17 | 21,058 | 6.6 | 0.22 | True | True | True | True |
| 2 | 95 | 40,793 | 6.7 | 0.48 | True | True | True | True |
| 3 | 473 | 222,232 | 7.5 | 1.27 | True | True | True | True |
| 4 | 2,159 | 359,041 | 8.0 | 2.32 | True | True | True | True |

5 Conclusion and Future Work

Many research proposals [6,10,13] have addressed the verification of behavior specifications in UML. Also, extensive research [2,9,11,14,16] has been done on the verification of multi-agent systems. However, few work focus on these two techniques together. This paper proposes a fully automated approach to verify agent-based communities of web services modeled in UML activity diagrams. We formalized UML activity diagrams using *agent hypergraphs* and specified properties using a new logic for agent communication: CTL^{*CA} . We defined and implemented the mapping rules for transforming the *agent hypergraphs* and CTL^{*CA} specifications into the ISPL language. Finally, we used MCMAS to experiment with the *PNAWS* protocol.

In this work, action formulae are only captured by the transition labels. Considering the full semantics of different actions in different situations is needed to check more complicated protocols. Our plan for future work is to extend CTL^{*CA} by adding different action formulae and proposing a new OBDD-based algorithm for the model checking. We also plan to extend MCMAS to be fully compatible with this new logic. Besides model checking, we are planning to use Model Driven Architecture (MDA), launched by the Object Management Group (OMG) in 2001 as a promising software design method, to develop a flexible

platform for agent-based communities of web services. We intent to use model transformation, which is a process that generates a refined model from a source model [8]. This process is based on a transformation definition, which is a set of transformation rules that describe how one or more constructs in the source language can be transformed into one or more constructs in the target language. This process can be achieve automatically, which helps in reducing programming errors and coding time.

References

1. Bentahar, J., Maamar, Z., Wan, W., Benslimane, D., Thiran, P., Subramanian, S.: Agent-based communities of web services: An argumentation-driven approach. *Service Oriented Computing and Applications* 2(4), 219–238 (2008)
2. Bentahar, J., Meyer, J.-J.C., Wan, W.: Model checking communicative agent-based systems. *Knowledge-Based Systems* 22(3), 142–159 (2009)
3. Bordini, R.H., Fisher, M., Pardavila, C., Wooldridge, M.: Model Checking AgentSpeak. In: *Proc. of the Int. J. Conf. on Autonomous Agents and Multiagent Systems*, pp. 409–416. ACM, Melbourne (2003)
4. Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., Tacchella, A.: NuSMV 2: An open source tool for symbolic model checking. In: Brinksmma, E., Larsen, K.G. (eds.) *CAV 2002*. LNCS, vol. 2404, pp. 359–364. Springer, Heidelberg (2002)
5. Clarke, E.M., Grumberg, O., Peled, D.: *Model checking*. MIT Press, Cambridge (1999)
6. Eshuis, R., Wieringa, R.: Tool support for verifying UML activity diagrams. *IEEE Trans. Software Eng.* 30(7), 437–447 (2004)
7. Halpern, J., Vardi, M.Y.: Model checking vs. theorem proving: a manifesto. In: *Proc. of the Int. Conf. on Principles of Knowledge Representation 1991*, pp. 325–334 (1991)
8. Kleppe, A., Warmer, J., Bast, W.: *MDA Explained, The Model-Driven Architecture Practice and Promise*. Addison Wesley, Boston (2003)
9. Lomuscio, A., Qu, H., Raimondi, F.: MCMAS: A model checker for the verification of multi-agent systems. In: Bouajjani, A., Maler, O. (eds.) *Computer Aided Verification*. LNCS, vol. 5643, pp. 682–688. Springer, Heidelberg (2009)
10. Planas, E., Cabot, J., Gomez, C.: Verifying action semantics specifications in UML behavioral models. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) *CAiSE 2009*. LNCS, vol. 5565, pp. 125–140. Springer, Heidelberg (2009)
11. Raimondi, F.: *Model Checking Multi-Agent Systems*. Ph.D. Thesis, University of London, London (2006)
12. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: Perram, J., Van de Velde, W. (eds.) *MAAMAW 1996*. LNCS (LNAI), vol. 1038, pp. 42–55. Springer, Heidelberg (1996)
13. Schafer, T., Knapp, A., Merz, S.: Model checking UML state machines and collaborations. *Elect. Notes in Theoretical Comp. Sc.* 55(3), 357–369 (2001)
14. van der Meyden, R., Gammie, P.: MCK: Model checking knowledge, <http://www.cse.unsw.edu.au/~mck/>
15. Weilkens, T.: *System Engineering with SysML/UML Modeling, Analysis, Design*. Morgan Kaufmann, Burlington (2007)
16. Wooldridge, M., Fisher, M., Huget, M., Parsons, S.: Model checking multi-agent systems with MABLE. In: *Proc. of the Int. J. Conf. on Autonomous Agents and Multi-Agent Systems*, pp. 952–959. ACM, New York (2002)

An Ambient Intelligent Agent Model Based on Behavioural Monitoring and Cognitive Analysis

Alexei Sharpanskykh and Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{sharp,treur}@few.vu.nl

Abstract. This paper proposes a way in which cognitive models can be exploited in practical applications in the context of Ambient Intelligence. A computational model is introduced in which a cognitive model that addresses some aspects of human functioning is taken as a point of departure. From this cognitive model relationships between cognitive states and behavioural aspects affected by these states are determined. Moreover, representation relations for cognitive states are derived, relating them to external events such as stimuli that can be monitored. Furthermore, by automatic verification of the representation relations on monitoring information the occurrence of cognitive states affecting the human behaviour is determined. In this way the computational model is able to analyse causes of behaviour.

Keywords: Autonomous agents, cognitive modelling, ambient intelligence.

1 Introduction

One of the interesting areas in which cognitive models can be applied in a practically useful manner is the area of Ambient Intelligence, addressing technology to contribute to personal care for safety, health and wellbeing; e.g., [1]. Such applications make use of sensor devices to acquire sensor information about humans and their functioning, and of intelligent devices exploiting knowledge for analysis of such information. Based on this, appropriate actions can be undertaken that improve the human's safety, health, and behaviour. Commonly, decisions about such actions are made by these intelligent devices based only on observed behavioural features of the human and her context (cf. [3]). A risk of such an approach is that the human is guided only at the level of her behaviour and not at the level of the underlying cognitive states causing the behaviour. Such a situation might lead to suggesting the human to suppress behaviour that is entailed by her internal cognitive states, without taking into account these cognitive states (and their causes) themselves.

As an alternative route, the approach put forward in this paper incorporates a cognitive analysis of the internal cognitive states underlying certain behavioural aspects. To this end, a computational model is described, in which a given cognitive model of the human's functioning is exploited. A cognitive model is formalised using the Temporal Trace Language (TTL) [2]. In contrast to many existing cognitive modelling approaches based on some form of production rule systems, TTL allows explicit representation of time and complex temporal relations.

By performing cognitive analysis the computational model is able to determine automatically which cognitive states relate to considered behavioural (or performance) aspects of the human, which external events (e.g., stimuli) are required to be monitored to identify these cognitive states (monitoring foci), and how to derive conclusions about the occurrence of cognitive states from such acquired monitoring information. More specifically, monitoring foci are determined by deriving representation relations for the human's cognitive states that play a role in the cognitive model considered. Within Philosophy of Mind a representation relation relates the occurrence of an internal cognitive state property of a human at some time point to the occurrence of other (e.g., external) state properties at the same or at different time points [7]. For example, the desire to go outside may be related to an earlier good weather observation. As temporal relations play an important role here, in the computational model these representation relations are expressed as temporal predicate logical specifications. From these temporal expressions externally observable events are derived that are to be monitored. From the monitoring information on these events the computational model verifies the representation expressions, and thus concludes whether the human is in such a state. Furthermore, in case an internal state has been identified that may affect the behaviour or performance of the human in a certain way, appropriate actions may be proposed.

The paper is organised as follows. The modelling approach is introduced in Section 2. An example used throughout the paper is described in Section 3. In Section 4 the proposed cognitive analysis approach is described. Finally, the paper is concluded with a discussion and summary.

2 Modelling Approach

To model the dynamics of cognitive processes with an indication of time, a suitable temporal language is required. In the current paper, to specify temporal relations the Temporal Trace Language (TTL) is used. This reified temporal predicate logical language supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. Dynamics are represented in TTL as an evolution of states over time. A state is characterized by a set of state properties expressed over (state) ontology Ont that hold. In TTL state properties are used as terms (denoting objects). To this end the state language is imported in TTL. Sort STATPROP contains names for all state formulae. The set of function symbols of TTL includes $\wedge, \vee, \rightarrow, \leftrightarrow$: $\text{STATPROP} \times \text{STATPROP} \rightarrow \text{STATPROP}$; not : $\text{STATPROP} \rightarrow \text{STATPROP}$, and \forall, \exists : $\text{S}^{\text{VARS}} \times \text{STATPROP} \rightarrow \text{STATPROP}$, of which the counterparts in the state language are Boolean propositional connectives and quantifiers. To represent dynamics of a system sort TIME (a set of time points) and the ordering relation $>$: $\text{TIME} \times \text{TIME}$ are introduced in TTL. To indicate that some state property holds at some time point the relation at : $\text{STATPROP} \times \text{TIME}$ is introduced. The terms of TTL are constructed by induction in a standard way from variables, constants and function symbols typed with all before-mentioned sorts. The language TTL has the semantics of many-sorted predicate logic. A special software environment has been developed for TTL, featuring a Property Editor for building TTL properties and a Checking Tool that enables automated formal verification of such properties against a set of traces.

The modelling approach presented in this paper adopts a rather general specification format for cognitive models that comprises past-present relationships between cognitive states and between cognitive states and sensor and effector states, formalised by temporal statements expressible within TTL. In this format, for a cognitive state a temporal pattern of past states can be specified, which causes the generation of this state; see also [6]. A *past-present statement* (abbreviated as a *pp-statement*) is a statement ϕ of the form $B \Leftrightarrow H$, where the formula H , called the *head* and denoted by $head(\phi)$, is a statement of the form $at(p, t)$ for some time point t and state property p , and B , called the *body* and denoted by $body(\phi)$, is a past statement for t . A *past statement* for a time point t over state ontology Ont is a temporal statement in TTL, such that each time variable s different from t is restricted to the time interval before t : for every time quantifier for a time variable s a restriction of the form $t > s$ is required within the statement. Sometimes B is called the *definition* of H .

Many types of cognitive models can be expressed in such a past-present format, such as causal models, dynamical system and connectionist models, rule-based models, and models in which memory of past events is used, such as case-based models. In the next section an example of a cognitive model specified in past-present format is given.

3 Case Study

To illustrate the proposed model a simplified example to support an elderly person in food and medicine intake is used. The following setting is considered. In normal circumstances the interval between two subsequent food intakes by the human during the day is known to be between 2 and 5 hours. When the human is hungry, she goes to the refrigerator and gets the food. Sometimes the human feels internal discomfort, which can be soothed by taking medicine X. The box with the medicine lies in a cupboard. There should be no food consumption for 2 hours after taking medicine. To maintain a satisfactory health condition of the human, intelligent support is employed, which is described by the computational model presented throughout the paper.

The behaviour of the human for this example is considered as goal-directed and is modelled using the BDI (Belief-Desire-Intention) architecture [9]. The graphical representation of the cognitive model that produces the human behaviour is given in Fig. 1. In this model the beliefs are based on the observations. For example based on

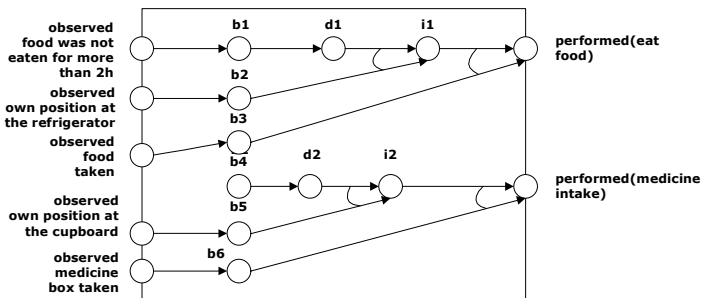


Fig. 1. Cognitive model for food and medicine intake

the observation that food is taken, the belief b_1 that food is taken is created. The desire and intention to have food are denoted by d_1 and i_1 correspondingly. The desire and intention to take medicine are denoted by d_2 and i_2 correspondingly. The model from the example was formalised by the following properties in past-present format:

IP1(c): General belief generation property

At any point in time a (persistent) belief state b about c holds *iff* at some time point in the past the human observed c . Formally: $\exists t_2 [t_1 > t_2 \ \& \ at(\text{observed}(c), t_2)] \Leftrightarrow at(b, t_1)$

IP2: Desire d_1 generation

At any point in time the internal state property d_1 holds *iff* at some time point in the past b_1 held. Formally: $\exists t_4 [t_3 > t_4 \ \& \ at(b_1, t_4)] \Leftrightarrow at(d_1, t_3)$

IP3: Intention i_1 generation

At any point in time the internal state property i_1 holds *iff* at some time point in the past b_2 and d_1 held. Formally: $\exists t_6 [t_5 > t_6 \ \& \ at(d_1, t_6) \ \& \ at(b_2, t_6)] \Leftrightarrow at(i_1, t_5)$

IP4: Action eat food generation

At any point in time the action eat food is performed *iff* at some time point in the past both b_3 and i_1 held. Formally: $\exists t_8 [t_7 > t_8 \ \& \ at(i_1, t_8) \ \& \ at(b_3, t_8)] \Leftrightarrow at(\text{performed}(\text{eat food}), t_7)$

IP5: Desire d_2 generation

At any point in time the internal state property d_2 holds *iff* at some time point in the past b_4 held. Formally: $\exists t_{10} [t_9 > t_{10} \ \& \ at(b_4, t_{10})] \Leftrightarrow at(d_2, t_9)$

IP6: Intention i_2 generation

At any point in time the action medicine intake is performed *iff* at some time point in the past b_5 and d_2 held. Formally: $\exists t_{12} [t_{11} > t_{12} \ \& \ at(d_2, t_{12}) \ \& \ at(b_5, t_{12})] \Leftrightarrow at(i_2, t_{11})$

IP7: Action medicine intake generation

At any point in time the action medicine intake is performed *iff* at some time point in the past both b_6 and i_2 held. Formally:

$\exists t_{14} [t_{13} > t_{14} \ \& \ at(i_2, t_{14}) \ \& \ at(b_6, t_{14})] \Leftrightarrow at(\text{performed}(\text{medicine intake}), t_{13})$

4 Cognitive Analysis

First, a set of goals is defined on the human's states and behaviour. The goal for the case study is to maintain a satisfactory health condition of the human. Each goal is refined into more specific criteria that should hold for the human's functioning. In particular, for the case study the goal is refined into three criteria:

- (1) food is consumed every 5 hours (at latest) during the day;
- (2) after the medicine is taken, no food consumption during the following 2 hours occurs;
- (3) after 3 hours from the last food intake no medicine intake occurs.

Based on the criteria expressions, a set of output states (called *an output focus*) and a set of internal (cognitive) states (called *an internal focus*) of the human are determined, which are used for establishing the satisfaction of the criteria. For the case study the output focus consists of the states $\text{performed}(\text{eat food})$ and $\text{performed}(\text{medicine intake})$.

A cognitive model of the human defines relations between an output state and internal states which cause the generation of the output state. The latter provide a more in depth understanding of why certain behaviours (may) occur. In general, using a cognitive model one can determine a minimal specification that comprises temporal relations to internal states, which provides necessary and sufficient conditions on

internal states to ensure the generation of an output state. An automated procedure to generate such specifications is considered in Section 4.1. Such a specification is a useful means for prediction of behaviour. That is, if an essential part of a specification becomes satisfied (e.g., when some important internal state(s) hold(s)), the possibility that the corresponding output state will be generated increases significantly. If such an output is (not) desired, actions can be proposed in a knowledgeable manner, based on an in depth understanding of the internal states causing the behaviour. Thus, the essential internal states (called *predictors for an output*) from specifications for the states in the output focus should be added to the internal focus.

Normally states in an internal focus cannot be observed directly. Therefore, representation relations are to be established between these states and externally observable states of the human (i.e., the representational content should be defined for each internal state in focus). Representation relations are derived from the cognitive model representation as shown in Section 4.2 and usually have the form of more complex temporal expressions over externally observable states. To detect occurrence of an internal state, the corresponding representational content should be monitored constantly, which is considered in Section 4.3.

4.1 Generating Predictors for Output States

A predictor(s) for a particular output can be identified based on a specification of human’s internal dynamics that ensures the generation of the output. In general, more than one specification can be identified, which is minimal in terms of numbers of internal states and relations between them, however sufficient for the generation of a particular output. Below an automated procedure for the identification of all possible minimal specifications for an output state based on a cognitive model is given. The rough idea underlying the procedure is the following. Suppose for a certain output state property p the pp-statement $B \Leftrightarrow at(p, t)$ is given. Moreover, suppose that in B only two atoms of the form $at(p1, t1)$ and $at(p2, t2)$ with internal states $p1$ and $p2$ occur, whereas as part of the cognitive model specifications $B1 \Leftrightarrow at(p1, t1)$ and $B2 \Leftrightarrow at(p2, t2)$ are available. Then, within B the atoms can be replaced (by substitution) by the formula $B1$ and $B2$. Thus, $at(p, t)$ may be related by equivalence to four specifications:

$$\begin{aligned}
 & B \Leftrightarrow at(p, t) && B[B2/at(p2, t2)] \Leftrightarrow at(p, t) \\
 & B[B1/at(p1, t1)] \Leftrightarrow at(p, t) && B[B1/at(p1, t1), B2/at(p2, t2)] \Leftrightarrow at(p, t)
 \end{aligned}$$

Here for any formula C the expression $C[x/y]$ denotes the formula C transformed by substituting x for y .

Algorithm. GENERATE-MINIMAL-SPECS-FOR-OUTPUT

Input: Cognitive model X ; output state in focus specified by $at(s, t)$

Output: All possible minimal specifications for $at(s, t)$ in list L

- 1 Let L be a list containing $at(s, t)$, and let δ_p, δ be empty substitution lists.
 - 2 For each formula $\varphi_i \in L$: $at(a_i, t) \leftrightarrow \psi_{i_p}(at_1, \dots, at_m)$ identify $\delta_i = \{at_k/body(\varphi_k) \text{ such that } \varphi_k \in X \text{ and } head(\varphi_k)=at_k\}$. Then δ is obtained as a union of δ_i for all formulae from L .
 - 3 $\delta = \delta \setminus \delta_p$
 - 4 if δ is empty, **finish**.
 - 5 For each formula $\varphi_i \in L$ obtain a set of formulae by all possible combinations of substitution elements from δ applied to φ_i . Add all identified sets to L .
 - 6 $\delta_p = \delta_p \cup \delta$, proceed to step 2.
-

For each generated specification the following measures can be calculated:

- (1) *The measure of desirability* indicating how desirable is the human's state, described by the generated specification at a given time point. The measure ranges from -1 (a highly undesirable state) to 1 (a highly desirable state).
- (2) *The minimum and maximum time before the generation of the output state(s)*. This measure is critical for timely intervention in human's activities.

These measures serve as heuristics for choosing one of the generated specifications. To facilitate the choice, constraints on the measures may be defined, which ensure that an intervention occurs only when a considerable (un)desirability degree of the human's state is determined, but also the minimum time before the (un)desirable output(s) is above some acceptable threshold. To calculate the measure (1), the degree of desirability is associated with each output state of the cognitive model. Then, it is determined which output states from the cognitive specification can be potentially generated, given that the bodies of the formulae from the generated specification are evaluated to TRUE. This is done by executing the cognitive specification with $\text{body}(\varphi_i) = \text{TRUE}$ for all φ_i from the generated specification. Then, the desirability of a candidate specification is calculated as the average over the degrees of desirability of the identified output states, which can be potentially generated. The measures (2) can be calculated when numerical timing relations are defined in the properties of a cognitive specification. After a specification is chosen, a set of predictor states from the specification for the output states in focus can be identified. When statistical information in the form of past traces of human behaviour is available, then the set of predictors is determined by identifying for each candidate two sets: a set of traces S in which the outputs in focus were generated and set $T \subseteq S$ in which the candidate set of predictors was generated. The closer the ratio $|T|/|S|$ to 1, the more reliable is the candidate set of predictors for the output(s) in focus.

For the case study from the automatically generated specifications that create the state *performed(eat food)* the one expressed by property IP4 is chosen. It has the desirability of the state *performed(eat food)*. Furthermore, it is assumed that the time interval $t7-t8$ in IP4 is sufficient for an intervention. The predictor state from the chosen specification is $i1$, as its predictive power depends on the occurrence of $b3$ only. Thus, $i1$ is included in the internal focus. By a similar line of reasoning, the specification expressed by property IP7 is chosen, in which $i2$ is the predictor state included into the internal focus. Thus, the internal focus for the cognitive model is the set $\{i1, i2\}$.

4.2 Representation Relations

A representation relation for an internal state property p relates the occurrence of p to a specification Φ that comprises a set of state properties and temporal relations between them. In such a case it is said that p *represents* Φ , or Φ describes *representational content* of p . In this section an automated approach to identify representation relations for cognitive states from a cognitive model is described.

The representational content considered backward in time is specified by a history (i.e., a specification that comprises temporal (or causal) relations on past states) that relates to the creation of some cognitive state. In the literature on Philosophy of Mind different approaches to defining representation relations have been put forward

(cf. [7]). For example, according to the classical causal/correlation approach, the representational content of an internal state property is given by a one-to-one mapping to an external state property. The application of this approach is limited to simple types of behaviour. In cases when an internal property represents a more complex temporal combination of state properties, other approaches have to be used. For example, the temporal-interactivist approach (cf. [6]) allows defining representation relations by referring to multiple (partially) temporally ordered interaction state properties; i.e., input (sensor) and output (effector) state properties over time.

To automate the representation relation identification based on this idea, a procedure was developed. To apply this procedure, cognitive specification is required to be stratified. This means that there is a partition of the specification $\Pi = \Pi_1 \cup \dots \cup \Pi_n$ into disjoint subsets such that the following condition holds: for $i > 1$: if a subformula $at(\varphi, t)$ occurs in a body of a statement in Π_i , then it has a definition within $\cup_{j < i} \Pi_j$.

Algorithm. GENERATE-REPRESENTATION-RELATION

Input: Cognitive specification X ; cognitive state specified by $at(s, t)$, for which the representation relation is to be identified

Output: Representation relation for $at(s, t)$

1 Stratify X :

1.1 Define the set of formulae of the first stratum ($h=1$) as $\{\varphi_i: at(a_i, t) \leftrightarrow \psi_{i_p}(at_1, \dots, at_m) \in X \mid \forall k \ m \geq k \geq 1 \ at_k \text{ is expressed using InputOnt}\}$; proceed with $h=2$.

1.2 The set of formulae for stratum h is identified as $\{\varphi_i: at(a_i, t) \leftrightarrow \psi_{i_p}(at_1, \dots, at_m) \in X \mid \forall k \ m \geq k \geq 1 \ \exists l \ l < h \ \exists \psi \in \text{STRATUM}(X, l) \ \text{AND} \ \text{head}(\psi) = at_k \ \text{AND} \ \exists j \ m \geq j \geq 1 \ \exists \xi \in \text{STRATUM}(X, h-1) \ \text{AND} \ \text{head}(\xi) = at_j\}$; proceed with $h=h+1$.

1.3 Until a formula of X exists not allocated to a stratum, perform 1.2.

2 Create the stratified specification X' by selecting from X only the formulae of the strata with the number $i < k$, where k is the number of the stratum, in which $at(s, t)$ is defined. Add the definition of $at(s, t)$ from X to X' .

3 Replace each formula of the highest stratum n of X' $\varphi_i: at(a_i, t) \leftrightarrow \psi_{i_p}(at_1, \dots, at_m)$ by $\varphi_i \ \delta$ with renaming of temporal variables if required, where $\delta = \{at_k\}$ body(φ_k) such that $\varphi_k \in X'$ and $\text{head}(\varphi_k) = at_k$. Further, remove all formulae $\{\varphi \in \text{STRATUM}(X', n-1) \mid \exists \psi \in \text{STRATUM}(X', n) \ \text{AND} \ \text{head}(\varphi) \text{ is a subformula of the body}(\psi)\}$

4 Append the formulae of stratum n to stratum $n-1$, which becomes the highest stratum ($n=n-1$).

5 Until $n > 1$, perform steps 3 and 4. The obtained specification with one stratum ($n=1$) is the representation relation specification for $at(s, t)$

In Step 3 subformulae of each formula of the highest stratum n of X' are replaced by their definitions, provided in lower strata. Then, the formulae of $n-1$ stratum used for the replacement are eliminated from X' . As result of such a replacement and elimination, X' contains $n-1$ strata (Step 4). Steps 3 and 4 are performed until X' contains one stratum only. In this case X' consists of a formula φ defining the representational content for $at(s, t)$, i.e., $\text{head}(\varphi)$ is $at(s, t)$ and $\text{body}(\varphi)$ is a formula expressed over interaction states and (temporal) relations between them.

In the following it is shown how this algorithm is applied for identifying the representational content for state it_1 from the internal focus from the case study. By performing Step 1 the specification of the cognitive model is automatically stratified: stratum 1: $\{IP1(\text{own_position_refrigerator}), IP1(\text{food_not_eaten_more_than_2h}), IP1(\text{own_position_cupboard}), IP1(\text{medicine_box_taken})\}$; stratum 2: $\{IP2, IP5\}$; stratum 3: $\{IP3, IP6\}$; stratum 4: $\{IP4, IP7\}$. By Step 2 the properties $IP4, IP5, IP6,$

IP7 are eliminated as unnecessary for determining the representational content of $i1$. In Step 3 we proceed with property IP3 of the highest stratum (3):

$$\exists t6 [t5 > t6 \ \& \ at(d1, t6) \ \& \ at(b2, t6)] \Leftrightarrow at(i1, t5)$$

In this step property IP8 is obtained by replacing $d1$ and $b2$ state properties in IP3 by their definitions with renaming of temporal variables:

$$\exists t6 [t5 > t6 \ \& \ \exists t4 [t6 > t4 \ \& \ at(b1, t4)] \ \& \ \exists t2 [t6 > t2 \ \& \ at(owned(own_position_refrigerator), t2)]] \Leftrightarrow at(i1, t5)$$

Further, the properties IP3, IP2 and IP1($owned_position_refrigerator$) are removed from the specification and the property IP8 is added to the stratum 2. Then, IP9 is obtained by replacing $b1$ in IP8 by its definition:

$$\exists t6 [t5 > t6 \ \& \ \exists t4 [t6 > t4 \ \& \ \exists t15 [t4 > t15 \ \& \ at(owned(food_not_eaten_more_than_2h), t15)]] \ \& \ \exists t2 [t6 > t2 \ \& \ at(owned(own_position_refrigerator), t2)]] \Leftrightarrow at(i1, t5)$$

After that properties IP8 and IP1($food_not_eaten_more_than_2h$) are removed from the specification and IP9 becomes the only property of the stratum 1. Thus, IP9 defines the representational content for the state $i1$ that occurs at any time point $t5$.

Similarly, the representational content for the other state from the internal focus $i2$ is identified as:

$$\exists t12 [t11 > t12 \ \& \ \exists t16 [t12 > t16 \ \& \ at(owned(own_position_cupboard), t16)]] \Leftrightarrow at(i2, t11)$$

The algorithm has been implemented in Java with the overall time complexity for the worst case is $O(|X|^2)$, where $|X|$ is the length of a cognitive specification X .

4.3 Behavioural Monitoring

To support the monitoring process, it is useful to decompose a representational content expression into atomic subformulae that describe particular interaction and world events. The subformulae are determined in a top-down manner, following the nested structure of the overall formula:

$$\begin{aligned} \text{monitor_focus}(F) &\rightarrow \text{in_focus}(F) \\ \text{in_focus}(E) \wedge \text{is_composed_of}(E,C,E1,E2) &\rightarrow \text{in_focus}(E1) \wedge \text{in_focus}(E2) \end{aligned}$$

Here $\text{is_composed_of}(E,C,E1,E2)$ indicates that E is an expression obtained from subexpressions $E1$ and $E2$ by a logical operator C (i.e., and, or, implies, not, forall, exists). At each decomposition step subexpressions representing events are added to the list of foci that are used for monitoring. This list augmented by the foci on the states from the output focus is used for monitoring. For the case study from the representation content for $i1$ and $i2$ atomic monitoring foci: $owned(food_not_eaten_more_than_2h)$, $owned(own_position_refrigerator)$ and $owned(own_position_cupboard)$ were derived.

Furthermore, the information on the states in the output and internal foci, on the chosen predictors for the output states, and on the identified representation relations is used to monitor constantly. As soon as an event from the atomic monitoring foci occurs, the component initiates automated verification of the corresponding representational content property on the history of the events in focus occurred so far. The automatic verification is performed using the TTL Checker tool (for the details on the verification algorithm see [2]).

Another task is to ensure that the goal criteria hold. The satisfaction of the criteria is checked using the TTL Checker tool. To prevent the violation of a criterion

promptly, information related to the prediction of behaviour (i.e., predictors for outputs) can be used. More specifically, if the internal states-predictors for a set of output states O hold, and some behaviour or performance criterion is violated under O , then an intervention in human activities is required. The type of intervention may be defined separately for each criterion. For the case study as soon as the occurrence of the prediction states i_1 and i_2 is established, the violation of the criteria identified previously is determined under the condition that the predicted outputs hold. To prevent the violation of the criteria, the following intervention rules are specified:

- (1) If the human did not consume food during last 5 hours, then inform the human about the necessary food intake.
- (2) If the human took medicine X less than 2 hours ago (time point t_2 in minutes) and the existence of the predictor i_1 is established, then inform the human that she still needs to wait $(120 - t_2)$ minutes for taking medicine.
- (3) If the human did not consume food during last 3 hours and the existence of the predictor i_2 is established, inform the human that she better eats first.

5 Discussion and Conclusions

In this paper a computational model was presented incorporating a more in depth analysis based on a cognitive model of a human's functioning. Having such a cognitive model allows relating certain behavioural or performance aspects that are considered, to underlying cognitive states causing these aspects. Often cognitive models are used either by performing simulation, or by temporal reasoning methods; e.g. [8]. In this paper a third way of using such models is introduced, namely by deriving more indirect relations from these models. Such an approach can be viewed as a form of knowledge compilation [4] in a pre-processing phase, so that the main processing phase is less intensive from the computational point of view. Such a form of automated knowledge compilation occurs in two ways: first, to derive the relationships between considered behaviour or performance aspects to the relevant internal cognitive states, and next to relate such cognitive states to observable events (monitoring foci). These monitoring foci are determined from the cognitive model by automatically deriving representation relations for cognitive states in the form of temporal specifications. From these temporal expressions the events are derived that are to be monitored, and from the monitoring information on these events the representation expressions are verified automatically.

A wide range of existing ambient intelligence applications is formalised using production rules (cf. [5]) and if-then statements. Two important advantages of such rules are modelling simplicity and executability. However, such formalism is not suitable for expressing more sophisticated forms of temporal relations, which can be specified using the TTL language. In particular, references to multiple time points possible in TTL are necessary for modelling forms of behaviour more complex than stimulus-response (e.g., to refer to memory states). Furthermore, TTL allows representing temporal intervals and to refer to histories of states, for example to express that a medicine improves the health condition of a patient.

Another popular approach to formalise recognition and prediction of human behaviour is by Hidden Markov Models (HMM) (e.g., [10]). In HMM-based approaches known to the authors, recognition of human activities is based on contextual information

of the activity execution only; no cognitive or (gradual) preparation states that precede actual execution of activities are considered. As indicated in [10] a choice of relevant contextual variables for HMMs is not simple and every additional variable causes a significant increase in the complexity of the recognition algorithm. Knowledge of cognitive dynamics that causes particular behaviour would provide more justification and support for the choice of variables relevant for this behaviour. Furthermore, as pointed in [3], for high quality behaviour recognition a large corpus of training data is needed. The computational costs of the pre-processing (knowledge compilation) phase of our approach are much lower (polynomial in the size of the specification). Also, no model training is required. However, the proposed approach relies heavily on the validity of cognitive models.

Acknowledgments. This work is supported under the FP7 ICT Future Enabling Technologies programme of the European Commission under grant agreement No 231288 (SOCIONICAL).

References

1. Aarts, E., Harwig, R., Schuurmans, M.: Ambient Intelligence. In: Denning, P. (ed.) *The Invisible Future*, pp. 235–250. McGraw Hill, New York (2001)
2. Bosse, T., Jonker, C.M., Meij, L., van der Sharpanskykh, A., Treur, J.: Specification and Verification of Dynamics in Agent Models. *Int. J. of Cooperative Information Systems* 18(1), 167–193 (2009)
3. Brdiczka, O., Langet, M., Maisonnasse, J., Crowley, J.L.: Detecting human behavior models from multimodal observation in a smart home. In: *IEEE Transactions on Automation Science and Engineering*, vol. 6(3). IEEE Computer Society Press, Los Alamitos (2009)
4. Cadoli, M., Donini, F.M.: A Survey on Knowledge Compilation. *AI Communications* 10(3-4), 137–150 (1997)
5. Christensen, H.B.: Using Logic Programming to Detect Activities in Pervasive Healthcare. In: Stuckey, P.J. (ed.) *ICLP 2002*. LNCS, vol. 2401, pp. 421–436. Springer, Heidelberg (2002)
6. Jonker, C.M., Treur, J.: A Temporal-Interactivist Perspective on the Dynamics of Mental States. *Cognitive Systems Research Journal* 4, 137–155 (2003)
7. Kim, J.: *Philosophy of Mind*. Westview Press (1996)
8. Port, R.F., van Gelder, T. (eds.): *Mind as Motion: Explorations in the Dynamics of Cognition*. MIT Press, Cambridge (1995)
9. Rao, A., Georgeff, M.P.: Modeling rational agents within a bdi-architecture. In: *Proc. of 2th Int. Conf. on Principles of Knowledge Repr. and Reasoning*, pp. 473–484 (1991)
10. Sanchez, D., Tentori, M., Favela, J.: Hidden Markov Models for Activity Recognition in Ambient Intelligence Environments. In: *Proc. 8th Int. Mexican Conf. Current Trends in Computer Science*, pp. 33–40. IEEE CS Press, Los Alamitos (2007)

The Combination of a Causal and Emotional Learning Mechanism for an Improved Cognitive Tutoring Agent

Usef Faghihi¹, Philippe Fouriner-Viger¹, Roger Nkambou¹, and Pierre Poirier²

¹ Department of Computer Science, UQAM
201, avenue du Président-Kennedy, Local PK 4150
Montréal, Québec, Canada

² Cognitive Science Institute, UQAM
Université du Québec à Montréal
C.P. 8888 succ. Centre-ville
Montréal, Québec, Canada, H3C 3P8

{Usef.faghihi, PhilippeFouriner-viger, Roger.Nkambou,
Pierre.Poirier}@courrier.uqam.ca

Abstract. This paper describes a Conscious Tutoring System (CTS) capable of dynamic fine-tuned assistance to users. We put forth the combination of a Causal Learning and Emotional learning mechanism within CTS that will allow it to first establish, through data mining algorithms, gross user group models. CTS then uses these models to find the cause of mistakes made by users, evaluate their performance, predict their future behavior, and, through a Pedagogical Knowledge mechanism, decide which tutoring intervention fits best.

Keywords: Autonomous Agents, Cognitive Tutoring Agent, Episodic Memory, Emotions, Causal Learning.

1 Introduction

CTS [1] is a cognitive agent designed to provide assistance during training in virtual learning environments. In this work, it is applied to a tutoring system in order to provide assistance to astronauts learning how to manipulate Canadarm2, the robotic telemanipulator attached to the International Space Station (ISS). CTS is partly based on the latest neurobiology and neuropsychology theories of human brain function (see Figure 1) and operates through cognitive cycles (five per second) which are based on LIDA [2]. The learners' manipulations of the virtual world simulator, simulating Canadarm2 [3], constitute the interactions between them and CTS. In particular, the virtual world simulator sends all manipulation data to CTS, which, in turn, sends learners various advices to improve their performance (Figure 2). One of CTS' most significant limitations, in its current implementation, is its incapacity to find out why an astronaut made a mistake, i.e, to find the causes of the mistakes. To address this issue, we propose to implement a Causal Learning Mechanism in CTS and combine it with its existing Emotional Learning Mechanism (see (Faghihi et al., 2008) for more details). In humans, the process of inductive reasoning stems in part from activity in the left prefrontal cortex and the amygdala; it is a multimodular process [4]. We base

our proposed improvements to CTS' architecture on this same logic. CTS' modular and distributed organization is ideal for the use of distinct mathematical methods and algorithms that can be tailored to the specific requirements of the Emotional Learning mechanism and the newly integrated Causal Learning mechanism.

Causal learning is the process through which we come to infer and memorize an event's reasons or causes based on previous beliefs and current experience that either confirm or invalidate previous beliefs [5]. In the context of CTS, we refer to Causal Learning as the use of inductive reasoning to generalize rules from sets of experiences (Purves et al., 2008). CTS observes astronaut behavior without complete information regarding the reasons for their behavior. Our prediction is that, through inductive reasoning, it can infer the proper set of causal relations from its observations of the astronaut's behavior.

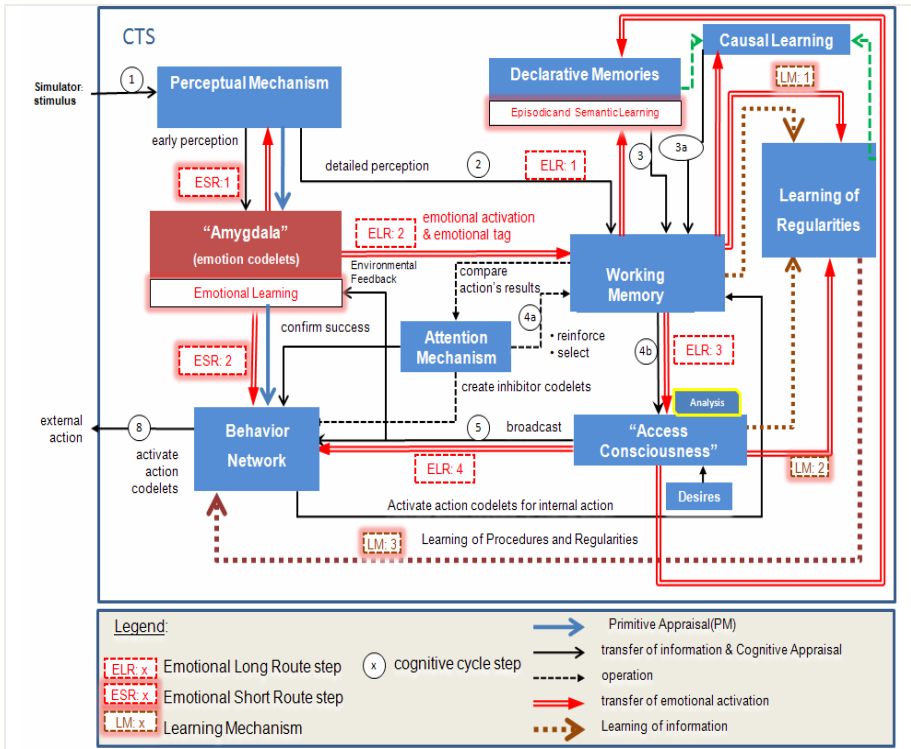


Fig. 1. CTS' Architecture

The goal of CTS' Causal Learning Mechanism (CLM) is two-fold: 1) to find causal relations between events during training sessions in order to better assist users; 2) to implement partial procedural learning in CTS' Behavior Network (BN) which is based on [6]¹. To implement CTS' CLM, we draw inspiration from Maldonado's

¹ CTS' BN (Figure 2.D) is a high-level procedural memory, a network of partial plans that analyses the context to decide what to do, which behavior to set off.

work [5], which defines three hierarchical levels of causal learning: 1) the lower level, responsible for the memorization of task execution; 2) the middle level, responsible for the computation of retrieved information; 3) the higher level, responsible for the integration of this evidence with previous causal knowledge.

In the present paper, we begin with a brief review of the existing work concerning the implementation of Causal Learning in cognitive agents. We then propose our new architecture combining elements of the Emotional Mechanism (EM) and Causal Learning. Finally, we present results from our experiments with this cognitive agent.

2 Causal Learning Models and Their Implementation in Cognitive Agents

To our knowledge, two cognitive research groups have attempted to incorporate Causal Learning mechanisms in their cognitive architecture. The first is Schoppek with the ACT-R architecture [7], who hadn't included a role for emotions in this causal learning and retrieval processes. ACT-R constructs the majority of its information according to the I/O knowledge base method. It also uses a sub-symbolic form of knowledge to produce associations between events. As explained by Schoppek [8], in ACT-R, sub-symbolic knowledge applies its influence through activation processes. However, the causal model created by Schoppek in ACT-R "*overestimates discrimination between old and new states*". The second is Sun [9] who proposed the CLARION architecture. In CLARION's current version, during bottom-up learning, the propositions (premises and actions) are already present in top level (explicit) modules before the learning process starts, and only the links between these nodes emerges from the implicit level (rules). Thus, there is no unsupervised causal learning for the new rules created in CLARION [10]. Various causal learning models have been proposed, such as Gopnik's model [11]. All proposed model use a Bayesian approach for the construction of knowledge. A problem arises, then, for Bayesian networks need experts to assign predefined values to variables. The second problem in Bayesian networks are the risk for combinatory explosion in the case of huge amount of data. In our case, according to the huge amount of data stores in CTS' modules, due to the interaction with learners, we argue that a combination of sequential pattern mining algorithms with association rules is more appropriate. The other advantage of causal learning using association rules is that the system learns in an incremental and real time manner- the system updates its information by interacting with various users. And finally, the aforementioned problem explained by Schoppek, and which occurs with ACT-R, cannot occur when using association rules for causal learning.

3 Causal Memory in CTS Architecture

CTS architecture relies on the functional "*consciousness*"²[2] mechanism for much of its operations. It also bears some functional similarities with the physiology of the

² Consciousness: Conscious cognition is implemented computationally by way of a broadcast of contents from a "global workspace", which receives input from the senses and from memory (Franklin & Patterson.2006).

nervous system. Its modules communicate with one another by contributing information to its Working Memory (WM) through information codelets³ [12] (see [1] for more details). CTS is equipped with an Emotional Mechanism that is based on the OCC (Ortony, Clore, & Collins, 1988) emotion model which extends from current models by defining learning in the emotional mechanism as one that helps different types of learning (e.g. Episodic Learning) and differentiating a variety of emotions (Faghihi et al., 2008). It is also equipped with an episodic mechanism that helps astronauts while they manipulate Canadarm2 in the virtual world. CTS' episodic mechanism collaborates with the emotional mechanism for the encoding and remembering of information [13]. In this work, we incorporate in CTS a form of self-sufficient learning - Causal Learning. This takes place through CTS' cognitive cycle (Figure 1) of which we give a brief overview below.

The cycle begins with a perception and ends with an action. The collaboration between CTS' Emotional mechanism and Causal Learning can be very briefly described in the following manner: CTS' WM is monitored by various types of codelets, one of which are expectation codelets (see (Faghihi et al., 2008) for more details). If expectation codelets observe information coming in WM confirming that the behavior's expected result failed, then the failure brings CTS' emotional and attention mechanisms back to that information. First, emotional codelets observing WM send a portion of emotional valences that is sufficient to get CTS' attention to select the failed information and bring it back to its consciousness. Emotional codelets' influence remains present throughout the following cognitive cycles, until CTS finds a solution or has no remedy for the failure.

Hereafter, we briefly summarize each step in CTS' cognitive cycle and in *italics*, describe the influence of emotions (here EM) and/or of CLM For a visual of the same, please refer to Figure 1.

Step 1: The first stage of the cognitive cycle is to perceive the environment; that is, to recognize and interpret the stimulus (see [1] for more information).

Step 2: The percept enters Working Memory (WM). The percept is brought into WM as a network of information codelets that covers the many aspects of the situation (see [1] for more information).

In this step, if the received information is considered important or dangerous by EM, there will be a direct reaction from EM which primes an automatic behavior from BN [14].

CLM: CLM also inspects and fetches WM relevant information. Relevant traces from different memories are automatically retrieved. These will be sequences of events in the form of a list relevant to the new information. The sequences include the current event, its relevant rules and the residual information from previous cognitive cycles in WM. The retrieved traces contain codelet links with other codelets. Each time new information codelets enter WM, the memory traces are updated depending on the new links created between these traces and the new information codelets. Once information enriched CLM sends it back to the WM.

³ Based on Hofstadter et al.'s idea, a codelet is a very simple agent, "a small piece of code that is specialized for some comparatively simple task". Implementing Baars theory's simple processors, codelets do much of the processing in the architecture. In our case, each information codelet possesses an activation value and an emotional valence specific to each cognitive cycle.

Step 3: Memories are probed and other unconscious resources contribute. All these resources react to the last few consciousness broadcasts (internal processing may take more than one single cognitive cycle).

Step 4: Coalitions assemble. In the reasoning phase, coalitions of information are formed or enriched. Attention codelets join specific coalitions and help them compete with other coalitions toward entering "consciousness".

EM: Emotional codelets observe the WM's content, trying to detect and instill energy to codelets believed to require it and attach a corresponding emotional tag. As a result, emotions influence which information comes to consciousness, and modulate what will be explicitly memorized.

Step 5: The selected coalition is broadcast. The Attention mechanism spots the most energetic coalition in WM and submits it to the "access consciousness," which broadcasts it to the whole system. With this broadcast, any subsystem (appropriate module or team of codelets) that recognizes the information may react to it.

CLM: First, CLM retrieves the past frequently reappearing information, ignoring temporal part of them, best matching the current information resident in WM. This occurs by constantly extracting associated rules from the broadcasted information and the list of special events previously consolidated. Then, CLM eliminates the rules that do not meet the temporal ordering of events.

Steps 6 and 7: Here unconscious behavioral resources (action selection) are recruited. Among the modules that react to broadcasts is the Behavior Network (BN). BN plans actions and, by an emergent selection process, decides upon the most appropriate act to adopt. The selected Behavior then sends away the behavior codelets linked to it.

EM: 0020When CTS' BN starts a deliberation, for instance to build a plan, the plan is emotionally evaluated as it is built, the emotions playing a role in the selection of the steps. If the looping concerns the evaluation of a hypothesis, it gives it an emotional evaluation, perhaps from learned lessons from past experiences.

CLM: the extraction of the rules in step 5, may invoke a stream of behaviors related to the current event, with activation passing through the links between them Figure 2.D). At this point CLM wait for the CTS' decision making mechanism and CTS' episodic learning mechanism solution for the ongoing situation) [13]. Then, CLM puts its proposition as a solution to CTS' WM, if decision making and episodic learning mechanisms propositions are not energetic enough to be chosen by CTS attention.

Step 8: Action execution. Motor codelets stimulate the appropriate muscles or internal processes.

EM: Emotions influence the execution, for instance in the speed and the amplitude of the movements.

CLM: The stream of behaviors activated in the CTS' BN (step 7) may receive inhibitory energies, from CLM, for some of their special behaviors. This means, according to CTS' experiences, CLM may use a shortcut (eliminates some intermediate nodes) between two nodes in behavior Network (BN) to achieve a goal (e.g., in Figure 2.D two points v and z). In some cases, again according to CTS' experiences, CLM may prevent the execution of unnecessary behaviors in CTS' BN during the execution of a stream of behaviors.

4 The Causal Learning Process

The next subsections give a detailed explanation of the three phases of the Causal Learning mechanism as it is implemented in CTS' architecture.

4.1 The Memory Consolidation Process

The causal memory consolidation process, which occurs in the Step 2 of CTS' cognitive cycle, takes place during each of CTS' cognitive cycles. Like the human left prefrontal cortex, CTS' CLM extracts past common events from its past experience, as they were recorded in its different memories. Events are information that CTS receives from Canadarm2 during astronauts' training sessions for arm manipulation [3] (Figure 2.A). A trace of what occurred in the system is recorded in CTS' different memories during consciousness broadcasts [13]. For instance, each event $X = (t_i, A_i)$ in CTS represents what happens during a cognitive cycle. While the timestamp t_i of an event indicates the cognitive cycle number, the set of items A_i of an event contains an item that represents the coalition of information codelets (see step 4 of CTS' cognitive cycle) that were broadcasted during the cognitive cycle. For example, one partial sequence recorded during our experimentations was $\langle (t=1, c2), (t=2, c4) \rangle$. This sequence shows that during cognitive cycle 1, the coalition $c2$ (that the user forgot to adjust the camera in the simulator, Figure 2.A) was broadcasted, followed by the broadcast of $c4$ (that the user made an imminent collision in the simulator, Figure 2.A) during cognitive cycle 2.

4.2 Learning Extracted Rules

The second phase of CLM occurs in Step 5 of CTS' cognitive cycle. First, it mines rules from the sequences of events by removing the time for each recorded event during CTS' executions. To do so, the algorithm takes as input the sequence database (sequences of coalitions that were broadcasted for each execution of CTS). It then produces the set of all causal rules contained in the database as output. The algorithm starts by ignoring the temporal information from the sequence database to obtain a transaction database. Once this is done, the algorithm then applies an association rule mining algorithm to discover all the association rules from this transaction database with a minimum support and confidence threshold defined by domain expert. It performs one pass on the original sequence database to eliminate the rules that do not meet the minimum support and confidence according to the temporal ordering of events, within a given time interval. The algorithm thus eliminates the non-causal rules. The set of rules that is kept will become the set of all causal rules [15].

4.3 Using Mined Patterns to Improve CTS' Behavior

The third part of Causal Learning, as explained above, occurs in Step 7 and Step 8 of CTS' cognitive cycle. It consists of improving CTS' behavior by making it reuse previous rules to anticipate the reasons for users' mistakes and to determine how to best help them.

5 Testing Causal Learning in the New CTS

To validate CTS' CLM, we integrated it into Canadarm2, our simulator designed to train astronauts to manipulate arm (**Figure 2.A**). Users were invited to perform arm manipulations using Canadarm2. In these experiments, users had to move the arm from one configuration to another in the simulator while avoiding collisions between the arm and the space station. This is a complex task, as the arm has seven joints and the user must (1) choose the best three cameras (from a set of about twelve cameras on the space station) for viewing the environment (since no camera offers a global view of the environment), (2) not move the arm too close to the ISS, (3) choose the right joint for the arm movements, and (4) adjust parameters of cameras properly. These experiments sought to validate CTS' ability to find the causes of mistakes made by users. During these experiments, we observed that CTS was able to find the causes and propose appropriate hints to help users. Some experiments are described next.

5.1 Users' Learning Situations

As previously mentioned, a user learns by practicing arm manipulations and receiving hints created initially by an expert and given to the user by CTS. The learner's success (defined as the extent of self-satisfaction in CTS) will be variable, depending on CTS' appropriate application of these hints.

We performed more than 250 CTS executions of arm in Canadarm2 including good moves and dangerous moves, such as collisions. During each execution, CTS chose a scenario depending on the situation. After each CTS execution, CLM extracted causal rules and the emotional valence attributed to the given scenario, and used these for future interactions. Our experiments showed that CTS is capable of finding the right causes of problems created by users in different situations. In what follows, two different experiments are detailed.

Scenario 1: Approximate problem

When manipulating Canadarm2 (arm), it is important for the users to know the exact distance between the arm and ISS at all times. This prevents future collisions or collision risks on ISS. **Figure 2.D** shows the scenario created by an expert in the CTS' Behavior Network (BN). This scenario is an intervention by CTS to help the user while manipulating the arm to avoid collisions between the arm and ISS. The user weakly estimated the distance between the arm and ISS because 1) the user chose to move the wrong joint; 2) the user was tired; 3) the user did not remember his course; 4) the user has never passed through this zone.

As we can see in the Figure 2.D, this is a long scenario and each time, to find the cause of mistakes made by the user, CTS may be required to interact for a long period of time (e.g. asking questions, giving hints and demonstrating some examples) to find the causes and provide appropriate feed-back. The scenario starts when CTS detects that a user has chosen the wrong joint and is moving the arm too close to the ISS. CTS first prompts the following message: Have you ever passed through this zone? 1) If the answer given by the user is *yes*, CTS asks the user to verify the name of the joint that he has selected. Then, if the user fails to answer correctly, CTS proposes a hint in the form of a demonstration or it stops arm manipulation. In this case, the user needs to revise the course before starting the arm manipulation again; 2) if the user's answer is *no*, CTS asks him to estimate the distance between the arm and ISS. If the user fails to answer correctly then the next hint from CTS asks the user if s/he is tired or forgot the course about this zone or if s/he needs some help; if the user answers correctly, it means that the user is an expert user and that the situation is not dangerous. Interacting with various users and according to the users' answers, CTS found the following rule 1) 60 % of the time "user chose the wrong joints \rightarrow user makes the arm pass too close to the ISS", Figure 2.D ($V \rightarrow W$); 2) in 35% of the time "user has never passed through this zone \rightarrow user manipulate near to the ISS", Figure 2.D ($V \rightarrow Z$); 3) in 5% of the time "user is an expert \rightarrow user makes the arm pass too close to the ISS", Figure 2.D ($V \rightarrow X$).

Scenario 2

It is a fact that users must perform Camera adjustments before moving the arm in the virtual world. During our experiments, we noted that this step was frequently forgotten by users, and moreover, users frequently did not realize that they had neglected this step, when asked to reflect back on their potential mistakes during their last performance. This increase collision risk (as depicted in Figure 2.A) in the virtual world. Interacting with various users and according to the users' answers, for this scenario, CTS found the following rule 1) in 60 % of the time "the user is tired \rightarrow the user performs a camera adjustment error"; 2) in 30 % of the time "the user forgot the course \rightarrow the user performs a camera adjustment error"; 3) in 10 % of the time "the user lacks motivation \rightarrow the user is inactive".

As mentioned above, after some experience, CTS' CLM is capable of inducing (by jumping from one point to another point in the BN, Figure 2.D) the source of the mistakes made by the users and propose a solution for them in the virtual world. However, given that CTS is a tutor and must interact with the user, jumping from start point to the end of the scenario (Figure 2.D, $V \rightarrow Z$) causes the elimination of some important steps in the BN. To prevent this, we tagged the important nodes in the BN as *not to be eliminated*. Thus after some experiments CLM, to go from $V \rightarrow Z$, obligatory, passed through intermediate nodes such as node *Y* (Figure 2.D). We call it the CTS' partial procedural learning (Step 8 of CTS' cognitive cycle).

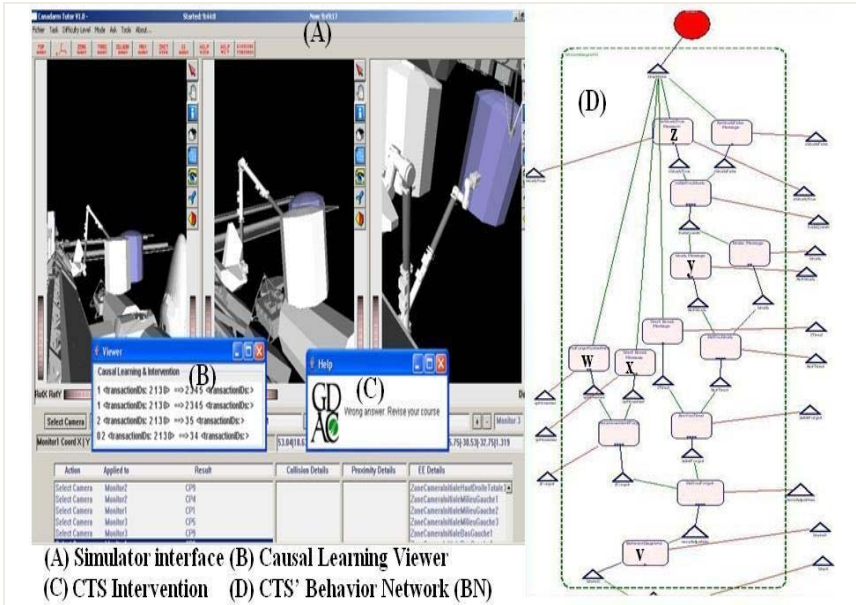


Fig. 2.

5.2 CTS' Performance after the Implementation of Causal Learning

We performed a second experiment with CTS' causal learning mechanism, but this time to observe how our association rule algorithm behaves when the number of recorded sequences increases. The experiment was done on a 3.6 GHz Pentium 4 computer running Windows XP, and consisted of performing more than 250 CTS executions for various situations (e.g., scenario 1 and scenario 2). In this situation, CTS conducts a dialogue with the user that includes from four to 20 messages or questions depending on what the user answers and the choices CTS makes. During each trial, we randomly answered the questions asked by CTS, and took various measures during CTS' learning phase. Each recorded sequence contained approximately 30 broadcasts. Figure 3 presents the results of the experiment. For all graphs, the X axis represents the executions from 1 to 250. The Y axis denotes execution times in graph A, and rule counts in graph B-D. The first graph (A) shows the time for mining rules which was generally short (less than 10 s) and after some executions remained low and stabilized at around 4 rules during the last executions. In our context, this performance is very satisfying. However, the performance of the rule mining algorithm could still be improved as we have not yet fully optimized all of its processes and data structures. In particular, in future works we will consider modifying the algorithm to perform incremental mining of rules. The second graph (B) shows number of causal rules found after each CTS execution. This would improve performance, at it would not be necessary to recompute from scratch the set of patterns for each new added sequence. The third graph (C) shows the average number of behaviors executed (nodes in the BN) for each CTS execution without causal learning. It ranges from 4 to 8 behavior broadcasts. The fourth graph (D) depicts, after the implementation of causal learning, the number of rules used by CTS at each

execution. Each executed rule means that CTS skipped some unnecessary intermediate steps in the BN. The average number of executed rules for each interaction ranged from 0 to 4 rules. This means that CTS generally used fewer nodes to perform the same task after the implementation of causal learning.

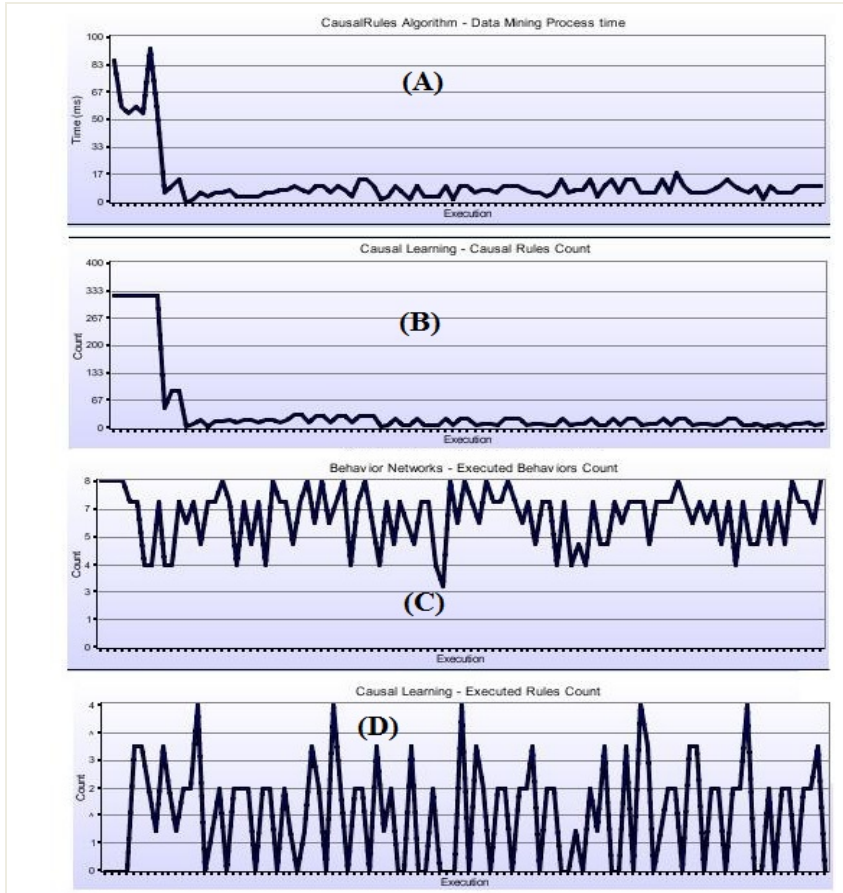


Fig. 3.

6 Conclusion

Reasoning is central to cognition. Scientists believe that humans first use heuristics or trial-and-error to solve problems. However, this approach is not useful for much of abstract reasoning, given its reliance of causal knowledge, which also supports our capacity for planning, imagination, inference, etc [4, 11]. In order to provide optimal tutoring assistance CTS must likewise be able to properly infer the causes of the users' mistakes in various situations. To our knowledge, researchers in artificial intelligence have up to now limited themselves to Bayesian methods to design causal reasoning and

causal learning models for cognitive agents. However, the Bayesian approach is not suitable when agents, such as CTS, face large amounts of data. This study, for the first time, combines sequential pattern mining algorithms and association rules to devise a causal learning model for a cognitive agent based on the sequential and temporal nature of the data stored in the system. Causal knowledge is generated in CTS after 1) the information is broadcasted in the system, 2) a decision made for the ongoing problem, which 3) is reinforced by future experiences while CTS interacts with its environment. The Emotional Learning mechanism is applied through the activation it sends to the information situated in CTS' working memory. This causes specific pieces of information to be chosen by CTS' Attention mechanism. This information, if mined by the causal learning algorithm, will more likely be activated in the future when CTS encounters similar problematic situations. Causal learning also helps partial procedural learning in CTS' Behavior Network (BN). After a certain number of similar experiences, the causal learning algorithm eliminates unnecessary nodes in CTS' BN. Our mechanism could then be considered as an alternative to a Bayesian algorithm. After implementing our Causal Learning Mechanism, we observed that CTS can find the causes of the users' mistakes in the virtual world and thus provide better tutoring assistance.

Acknowledgments. Our special thanks to Sioui Maldonado Bouchard for her collaboration in this paper. The authors also thank the Fonds Québécois de la Recherche sur la Nature et les Technologies.

References

- [1] Dubois, D., Poirier, P., Nkambou, R.: What Does Consciousness Bring to CTS? In: Woolf, B.P., Aimeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 803–806. Springer, Heidelberg (2008)
- [2] Franklin, S., Patterson, F.G.J.: The LIDA architecture: adding new modes of learning to an intelligent, autonomous, software agent. *Integrated Design and Process Technology* (2006)
- [3] Nkambou, R., Belghith, K., Kabanza, F., Khan, M.: Supporting Training on Canadarm Simulator using a Flexible Path Planner. In: *Artificial Intelligence in Education*, pp. 953–955. IOS Press, Amsterdam (2005)
- [4] Purves, D., Brannon, E., Cabeza, R., Huettel, S.A., LaBar, K., Platt, M., Woldorff, M.: *Principles of cognitive neuroscience*. Sinauer Associates, Sunderland (2008)
- [5] Maldonado, A., Catena, A., Perales, J.C., Cándido, A.: *Cognitive Biases in Human Causal Learning* (2007)
- [6] Maes, P.: How to do the right thing. *Connection Science* 1, 291–323 (1989)
- [7] Anderson, J.R.: *Rules of the mind*. Erlbaum, Mahwah (1993)
- [8] Schoppek, W.: Stochastic Independence between Recognition and Completion of Spatial Patterns as a Function of Causal Interpretation. In: *Proceedings of the 24th Annual Conference of the Cognitive Science Society* (2002)
- [9] Sun, R.: The CLARION cognitive architecture: Extending cognitive modeling to social simulation. In: *Cognition and Multi-Agent interaction*. Cambridge University Press, New York (2006)
- [10] Hélie, S.: "Modélisation de l'apprentissage ascendant des connaissances explicites dans une architecture cognitive hybride," PHD, DIC, UQAM, Montréal (2007)

- [11] Gopnik, A., Glymour, C., Sobel, D.M., Schulz, L.E., Kushnir, T., Danks, D.: A Theory of Causal Learning in Children: Causal Maps and Bayes Nets. *Psychological Review* 111(1) (2004)
- [12] Hofstadter, D.R., Mitchell, M.: The Copycat Project: A model of mental fluidity and analogy-making. In: Holyoak, K.J., Barnden, J.A. (eds.) *Advances in connectionist and neural computation theory*, Ablex. logical connections, vol. 2 (1994)
- [13] Faghihi, U., Fournier-Viger, P., Nkambou, R., Poirier, P., Mayers, A.: How Emotional Mechanism Helps Episodic Learning in a Cognitive Agent. In: *Proceedings of the 2009 IEEE Symposium on Intelligent Agents* (2009)
- [14] Faghihi, U., Poirier, P., Dubois, D., Nkambou, R.: Implementation of Emotional Learning for Cognitive Tutoring Agents. In: *7th Mexican International Conference on Artificial Intelligence (MICAI 2008)*. IEEE Computer Society press, Los Alamitos (2008)
- [15] Fournier-Viger, P., Faghihi, U., Nkambou, R., Mephu Nguifo, E.: CMRULES: An Efficient Algorithm for Mining Sequential Rules Common to Several Sequences. In: *FLAIRS Conference* (2010)

Driver's Behavior Assessment by On-board/Off-board Video Context Analysis

Lorenzo Ciardelli^{1,*}, Andrea Beoldo², Francesco Pasini¹, and Carlo Regazzoni¹

¹ Department of Biophysical and Electronic Engineering, University of Genoa
Via Opera Pia 11A, I-16145 Genoa, Italy

{ciardelli,pasini,carlo}@ginevra.dibe.unige.it

² TechnoAware s.r.l, Corso Buenos Aires 18/11, 16129 Genova, Italy
andrea.beoldo@technoaware.com

Abstract. In the last few years, the application of ICT technologies in automotive field has taken an increasing role in improving both the safety and the driving comfort. In this context, systems capable of determining the traffic situation and/or driver behavior through the analysis of signals from multiple sensors (e.g. radar, cameras, etc...) are the subject of active research in both industrial and academic sectors. The extraction of contextual information through the analysis of video streams captured by cameras can therefore have implications in many applications focused both on prevention of incidents and on provision of useful information to drivers. In this paper, we investigate the study and implementation of algorithms for the extraction of context data from on-board cameras mounted on vehicles. A camera is oriented so as to frame the portion of road in front of the vehicle while the other one is positioned inside the vehicle and pointed on the driver.

1 Introduction

The design of automatic systems for preventing driving incidents is a current and active domain of research involving both automobile manufacturers and academia. For these applications the analysis of what happens inside and outside a car is one of the most interesting sources of information.

As a matter of fact, the joint analysis of on board/off board car context can be used to derive considerations on driver's behavior and then to detect possible dangerous situations (sleep, dangerous lane changes, etc.) or a driving style which does not respect traffic regulation (see [1] and [18]).

1.1 On-board Analysis

The most significant data that can be extracted from a camera monitoring the driver are the gaze direction, the position of face, frequency of blinking eyes and mouth state. In the state of art several works can be found on face pose

* Corresponding author.

estimation, which is the feature we will focus on for on-board analysis, mainly addressed to Human-Computer Interaction or automotive applications. In [2] a complete and up-to date survey of head pose estimation algorithms is presented. Various approaches are categorized according to the typology of the approaches by pointing out the condition of usage, the assumptions and the obtainable performances. In [3] lips and eyes are located in the image tracked over time exploiting their color characteristic with respect to the skin. Then a 3D model is constructed to determine the gaze angles using constant projection assumptions. In [4] gaze is also estimated by analyzing the motion of some relevant features in the eyes and mouth area. This last method however does not take into account possible illumination changes since it is designed for indoor Human-Computer Interaction applications. The paper by Tu et al. [5] instead relies only on nose tip localization to estimate face pose by using "tensorposes" models. Other works try to cope with the difficult environmental conditions of the automotive applications (i.e. frequent and relevant illumination changes) by detecting eyes using infrared cameras (e.g. see [1] and [6]). These approaches are usually more robust and they can operate also with very low illumination but the cost of an infrared camera is much higher than a traditional webcam.

1.2 Off-board Analysis

The focus of this section is to detect the position of the vehicle on the road and the lane changes. Moreover, the analysis of the road type (highway, urban road, etc.) and of traffic is performed to provide relevant information to evaluate the possible risks of the driving behavior. In the literature, several works can be found addressing the problem of lane detection and tracking, however we will concentrate on lane detection using video sensors since they perform well in several situations. In [7], a survey of lane detection algorithms is proposed where the key element of these algorithms are outlined. A 360 degrees single PAL camera-based system is presented in [8], where authors provide both the driver's face pose and eye status and the driver's viewing scene basing on a machine learning algorithm for object tracking. A widely used technique to post-process of the output of the road marking extraction is the Hough transform as shown for example in [9]. In [10], a generic method for a probabilistic identification of driving situations and maneuvers was introduced basing on a Bayesian network and fuzzy features as input parameters. Such framework allows to identify emergency braking situations and lane changes with a good accuracy. In the paper by Wang et al. [11], a road detection and tracking method based on a condensation particle filter for real-time video-based navigation applications is presented. Worth of note is that most of these works are tested on highways where the background is usually less variable and complex with respect to urban roads.

The rest of the paper is organized as follows. In Section 2 the system architecture is briefly presented. In Section 3 on-board analysis approach is described as well as off-board analysis in Section 4 with some preliminary results. Finally in Section 5 conclusions are drawn and future developments are discussed.

2 System Architecture

The proposed architecture is composed by two general purpose webcams connected to a laptop computer installed inside the car. The PC has an internet connection through a UMTS card in order to make available context information as well as movies registered on the vehicle on a remote server.

A schematic representation of system physical and logical architectures is presented in Fig. 1 and 2. Modules 1 (TLC1) and 2 (TLC2) are the thread capture images from webcam using 2. The acquired images are sent to recorders that generate synchronized movies exploiting TIMER module which is a timer thread that sends to recorders signals concerning start / end of the movie.

Therefore, data acquisition and processing is carried out by equipping a car with a general purpose laptop linked with two webcams appropriately positioned to frame the exterior of car and the driver. Finally, a GUI has been developed for the acquisition and storage of various real-time synchronized data. The acquisition

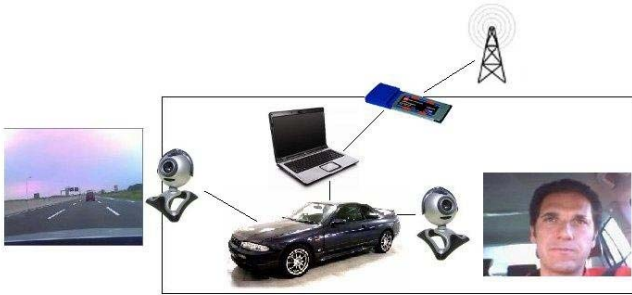


Fig. 1. System physical architecture

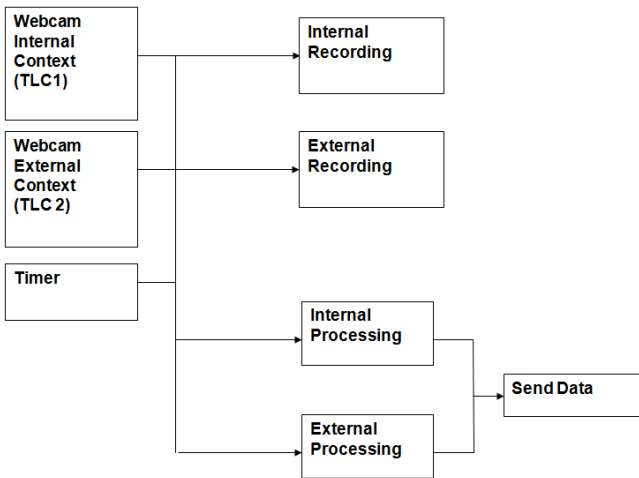


Fig. 2. System logical architecture

module allows to receive up to a maximum of 4 video streams at 25 frames/sec, also obtained by different cameras, and to timely synchronize data in order to consistently associate the information processed by each sensor.

3 On-board Context Analysis

In this work, a tracking based approach for driver gaze detection is proposed to the aim of obtaining a better ratio between accuracy and speed. Such method relies on the evaluation of the relative movement of the head between consecutive frames of a video sequence [17]. In the proposed method three processing step have been considered: (a) face detection (face, eyes, mouth and nose), (b) face tracking and (c) face analysis and angle of view calculation. Firstly, an initialization step is performed for face detection. Secondly, the tracking algorithm enables localizing the position of the face in the video frame and evaluating the relative position of every facial trait like the nose, the mouth and the eyes. Lastly, when face position estimation has been performed, for each video frame the pose of the face can be evaluated in order to extract the angle of view and other relevant information. In the next subsections detection, a more detailed description of the previously presented steps is provided.

3.1 Face Detection

For each trait (face, eyes, mouth, nose) Viola-Jones detector [15] is applied. This algorithm was broadly used in the last years for a lot of detection applications, in particular to the aim of localizing faces. Three major phases are performed: feature extraction, classification using boosting and multi-scale detection, enabling a fast and highly accurate detection. In such a method five detectors need to be used for identifying each face component leading to a computationally expensive approach that can hardly comply with required fast processing time. To reduce the computational effort, direct bottom-up detection has been substituted by a top-down geometrically constrained initialization.

3.2 Face Tracking

Face tracking cannot be consistently performed using a single detector because all detectors are sensible to rotations. According to this statement, for each face trait an instance of Kanade-Lucas-Tomasi (KLT) feature tracker algorithm [16] has been used. Feature selection has been specifically studied in order to maximize the quality of tracking, and it is therefore optimal by construction with respect to ad-hoc measures based on textures. Moreover, processing phase is computationally inexpensive and helps discriminating between good and bad features based on a measure of dissimilarity related to affine motion depending on the change of the underlying image model. This choice allows obtaining a good precision in terms of estimate of eyes position and, at the same time, ensures to keep eyes position information when great neck twists occur. Finally, KLT features allow extracting additional information on face rotation and shape changes that can be used to evaluate driver's pose.

3.3 Face Analysis

View angle is one of the most important information that is needed to assess to driver state. It can be disassembled in yaw (rotation with respect to horizontal plane), roll (longitudinal rotation related to movement) and pitch (vertical rotation) angles. In this paper, a rough but fast estimation of outlook direction is proposed. To this end, gaze angle has been subdivided in a set of n discrete possible values. To estimate yaw angle, KLT features position, the displacement of eyes during tracking phase and the triangle created by the two eyes and the nose are used. A first value ϵ of the yaw angle can be calculated form KLT feature as follows:

$$\epsilon = \frac{\text{var}(R_x)}{\text{var}(L_x)} \quad (1)$$

where $\text{var}(R_x)$ is the variance of right eye on x axis and $\text{var}(L_x)$ is the variance of left eye on x axis. It is important to point out that the relative movement of eyes' position during the tracking step together with the studies of the relative position between eyes and nose lead to an improvement of yaw angle estimation. If information related to nose position is not available or cannot be properly acquired, the triangle between the nose and the eyes cannot be created so only the information provided by KLT feature and eyes moving are used. In the latter case, the yaw angle evaluation could be affected; however preliminary results show that a good estimation can be obtained. On the other hand, roll angle evaluation has been performed basing on two features: (a) the relative position of the centers of the eyes with respect to the centre of the face and (b) the difference between the y coordinate values of each eye. As a general rule, we have assumed (having been demonstrated in a large testing phase) that values of the yaw angle near to 0 correspond to the situation of driver looking straight ahead (i.e. driver is looking at the street and his/her level of attention is adequate) while values far from 0 correspond to the case of driver looking in other directions rather than street one (i.e. a possible dangerous situation can happen because the driver is absent-minded).

3.4 Preliminary Results

A lot of experiments have been performed using a simple webcam at 320x240 of resolution. The webcam has been installed on a car and it has been used to analyze a driver during a thirty minutes drive. In the proposed experiments (Fig. 3) the red points indicate the eyes, yellow and green points indicate respectively the nose and the mouth. The purple line on the top of the face indicates the roll angle while yellow/blue one shows the direction of the yaw angle. Table 1 shows the experimental result obtained by the on-board video analysis. The percentage of frame with errors is obtained comparing algorithm results with observations. A more significant percentage of errors occur in detection and tracking phases rather than in angle view calculation.

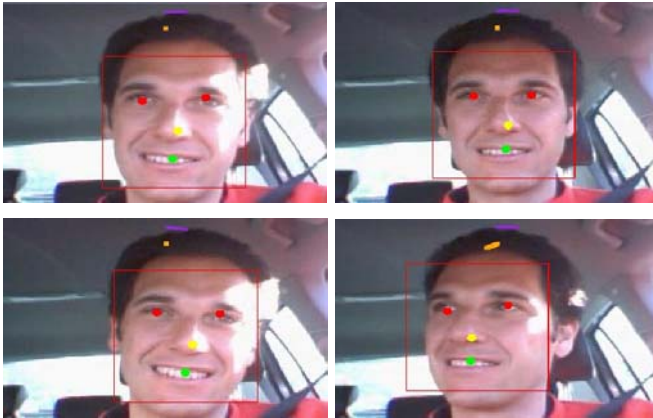


Fig. 3. On-board context analysis - preliminary results

Table 1. Percentage of frame with errors

| | |
|--------------------------|-------|
| No Face Detection | 3,1% |
| No Nose/Mouth Detection | 0,8% |
| Eyes/Nose Tracking Error | 11,4% |
| Mouth Tracking Error | 7,8% |
| Yaw Angle Error | 4,7% |
| Roll Angle Error | 16,1% |

This output can be justified taking into account that the continuous change of luminosity, which usually affects driving situations, either forces the proposed method to re-initialize the detection algorithm or leads to unexpected errors in face tracking.

4 Off-board Context Analysis

The purpose of the algorithms for off-board context analysis is to extract information about what is happening outside the vehicle. In particular, in order to evaluate the level of safety related to driver's behavior, the main features which will be considered are:

- number and position of roadways;
- position of the vehicle on the road with respect to traffic lines.

4.1 Method Implementation

The following steps have been applied to extract road context information from a video sequence:

1. edges extraction from each frame using the Canny operator [12] (Fig. 4(a)). The Canny operator is a very well known method in image processing to identify in the image the connected areas characterized by significant luminosity intensity difference with respect to neighbor pixels by computing the gradient of the image intensity value;
2. after extracting the edges, the resulting binary image is used to detect the lines using Hough algorithm [13] [14] (Fig. 4(b)). In this application a mask is applied to exclude areas of non-interest. The algorithm receives as input the coordinates of the points in the binary image and provides a parametric description of the recognized curves, belonging to a given analytic set (in our case the lines);

As it can be observed from Fig. (Fig. 4(b)) detected lines must be post-processed in order to select only those that are candidate to identify the lanes. To this end, under the assumptions that the camera acquires a frontal view of the road, the following steps are performed:

3. the two lines that belong to the lane where the vehicle is driving on are located. The closest line on the left with respect to the center of the image and the one on the right are considered as the first reference lines and they are identified respectively as l_0 and l_1 ;
4. once extracted the two lines, attention is focused on an area within the triangle formed by them. A frame per frame statistical analysis of the pixels belonging to the road is performed to create a model of the road. In particular, the color model of the road is evaluated to select the areas where to look for the other candidate lines. A single Gaussian probabilistic model $N(\mu, \Sigma)$ of the color of the road is learned computing the sample mean and sample variance on the three color channels red, blue and green, where μ is the mean and Σ is the covariance matrix;
5. all pixels in the image below the point of intersection between the two lines identified at step 3. are considered and each pixel is compared with the Gaussian model of the road looking for those that are more similar to the model. An example can be seen in Fig. 4(c);
6. the next step is to evaluate whether the road has one or two lanes and which is the position of the vehicle with respect to them. Considering all the lines extracted in step 3., one on the left l_{0L} and one on the right l_{1R} that are sufficiently far respectively from l_0 and l_1 , are considered. The fraction of road pixels with respect to all the ones within the two candidate lanes identified by $(l_0; l_{0L})$ and $(l_1; l_{1R})$ is computed. If this fraction is greater than a predefined value (e.g. 70%) and if the intersection point between l_0 and l_{0L} (or l_1 and l_{1R}) is close enough to the intersection point between l_0 and l_1 , than the candidate lane is considered as a valid one. An example of the final output of the lane detection procedure is shown in Fig. 4(d).

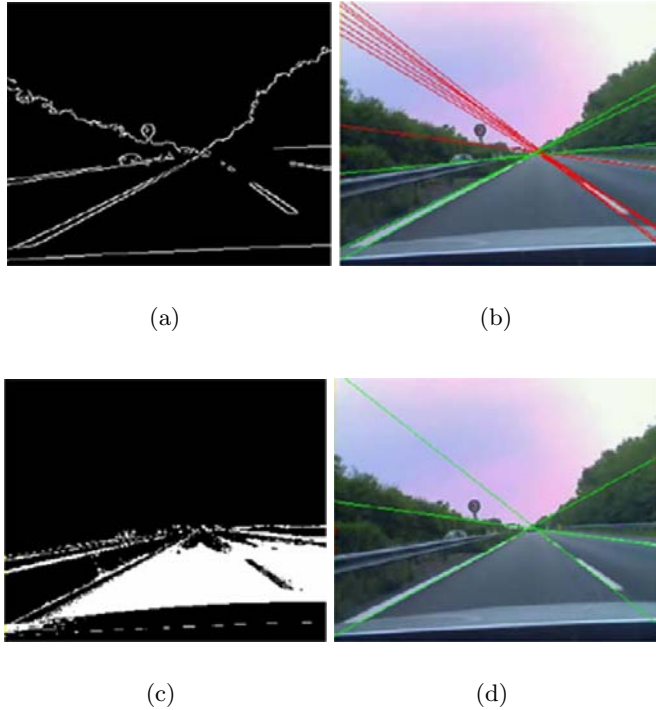


Fig. 4. a) Image of extracted edges; b) Lines detected using Hough transform algorithm; c) Road segmentation image; d) Extracted lines after elimination of not consistent lines

4.2 Preliminary Results

In order to test the proposed approach a webcam positioned on the vehicle dashboard has been used to acquire video sequence of real-world guiding scenes both in highways and urban road scenarios. During the tests, different contextual information has been considered:

1. Number of roadways;
2. Vehicle position.

Since both the number of roadways and the position of the vehicle does not instantly change algorithm output obtained for each frame of the video sequences have been averaged over a longer temporal window. More in detail, the number of lanes has been evaluated over a window of 2 seconds while for the analysis of the position of the vehicle a 1 second window has been used. In Figure 5 the GUI used for receiving and showing contextual information is presented while in Table 2 statistical data concerning vehicle's behavior are shown.



Fig. 5. Contextual data extraction GUI

Table 2. Percentage of frame with errors

| | Detections | Correct | Wrong | % Correct |
|--------------------|------------|---------|-------|-----------|
| Vehicle position | 30 | 25 | 5 | 83% |
| Number of roadways | 15 | 11 | 4 | 74% |

5 Conclusions and Future Work

In this work, a framework for analyzing driver's attention, detecting lanes and individuating vehicle position is proposed. This information can be relevant to design Intelligent Vehicles able to understand driver behavior and intent for preventive safety. Proposed methods are able to cope with typical difficulties present in this scenario such as illumination changes and dynamic background. Experiments performed on real world video sequences taken on-board cameras demonstrate the robustness of the presented framework and the capability to operate in a real-time fashion. From the one hand, future work will be focused on calculation of other parameters for identifying driver state, like evaluation pitch component of angle of view, analysis of blinking frequency of the eyes and a value of mouth state (open, close, speaking). To the other hand, traffic analysis and evaluation of other vehicles state and behavior will be explored to empower off-board contextual analysis.

References

1. Trivedi, M., Gandhi, T., McCall, J.: Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *IEEE Transactions on Intelligent Transportation Systems* 8(1), 108–120 (2007)
2. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 31(4), 607–626 (2009)
3. Smith, P., Shah, M., da Vitoria Lobo, N.: Determining driver visual attention with one camera. *IEEE Transactions on Intelligent Transportation Systems* 4(4), 205–218 (2003)

4. Asteriadis, S., Tzouveli, P., Karpouzis, K., Kollias, S.: Estimation of behavioral user state based on eye gaze and head pose-application in an e-learning environment. *Multimedia Tools Appl.* 41(3), 469–493 (2009)
5. Tu, J., Fu, Y., Huang, T.S.: Locating nose-tips and estimating head poses in images by tensorposes. *IEEE Transaction on Circuits and Systems for Video Technology* 19(1) (2009)
6. Bergasa, L.M., Nuevo, J., Sotelo, M., Barea, R., Lopez Guillen, M.L.: Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems* 7(1), 63–77 (2006)
7. McCall, J.C., Trivedi, M.M.: Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE Transaction on Intelligent Transportation Systems* 7(1), 20–37 (2006)
8. Yu, G., Xiao, X., Bai, J.: Analysis of vehicle surroundings and driver status from video stream based on a single PAL camera. In: 9th International Conference on Electronic Measurement and Instruments, 2009. ICEMI 2009, August 16–19, pp. 4-363 – 4-367 (2009)
9. Voisin, V., Avila, M., Emile, B., Begot, S., Bardet, J.-C.: Road markings detection and tracking using hough transform and kalman filter. In: Blanc-Talon, J., Philips, W., Popescu, D.C., Scheunders, P. (eds.) ACIVS 2005. LNCS, vol. 3708, pp. 76–83. Springer, Heidelberg (2005)
10. Schneider, J., Wilde, A., Naab, K.: Probabilistic approach for modeling and identifying driving situations. In: *IEEE Intelligent Vehicles Symposium*, June 4-6, pp. 343–348 (2008)
11. Wang, Y., Bai, L., Fairhurst, M.: Robust Road Modeling and Tracking Using Condensation. *IEEE Transactions on Intelligent Transportation Systems* 9(4), 570–579 (2008)
12. Canny, J.: A Computational Approach To Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence* 8, 679–714 (1986)
13. Hough, P.V.C.: Machine Analysis of Bubble Chamber Pictures. In: *Proc. Int. Conf. High Energy Accelerators and Instrumentation* (1959)
14. Duda, R.O., Hart, P.E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Comm. ACM* 15, 11–15 (1972)
15. Viola, P., Jones, M.: Robust real-time object detection. *International Journal of Computer Vision* (2002)
16. Shi, J., Tomasi, C.: Good features to track. *Computer Vision and Pattern Recognition*. In: *Proceedings of IEEE Computer Society Conference on CVPR 1994*, pp. 593–600 (1994)
17. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. *ACM Comput. Surv.* 38 (2006)
18. Oliver, N., Pentland, A.: Driver Behavior Recognition and Prediction in SmartCar. In: *Proceedings of SPIE Aerosense2000 'Enhanced and Synthetic Vision'*, Orlando, Florida (April 2000)

An eHealth System for a Complete Home Assistance

Jaime Martín¹, Mario Ibañez¹, Natividad Martínez Madrid¹, and Ralf Seepold²

¹ Universidad Carlos III de Madrid

{jaime.martin,mario.ibanez,natividad.martinez}@uc3m.es

² Hochschule Konstanz (HTWG), University of Applied Sciences

ralf.seepold@htwg-konstanz.de

Abstract. Home telecare systems are improving the current level of quality in healthcare services. This paper describes an eHealth system designed to support people living in their homes. The approach introduces a flexible system architecture that is running on a common residential gateway. The architecture provides basic services and openness to integrate dedicated telecare services. Special attention is paid to the integration of the patient's relatives and friends. For that purpose, a videoconference system allows any participant to show information about the availability, current status, to communicate face-to-face or in a discussion together with other members (e.g. patient, nurse, doctor and relatives). A module to maintain medical appointments has been integrated as well.

Keywords: eHealth, telemedicine, telecare, home healthcare, elderly people, HL7, OSGi, UPnP.

1 Introduction

The Prague Declaration [1], adopted during the eHealth European Ministerial Conference celebrated in Prague in February 2009, shows the relevance that eHealth has nowadays and in the future. This declaration presents the different stakeholders involved in the development of eHealth, and it highlights as well that “the lack of interoperability has been identified as one of the main areas to address”.

The main target group of telecare and telemedicine (in this approach) are people chronically ill, elderly or handicapped. Telecare utilizes information and communication technologies to transfer medical information for diagnosis and therapy of patients in their place of domicile while telemedicine is related to the delivery of clinical care at distance [2], for example a teletransmission of ECG (electrocardiograph).

Telecare services can significantly increase the quality of life for this group of people. However, there is still a lack in standardization that would allow to connect and to maintain the equipment provided from different vendors in a compatible and reliable way. Beyond the pure technical aspect, the incorporation of persons forming part of daily life is a crucial point.

Figure 1 shows a scenario that presents all stakeholders incorporated in our work. This includes the patient, relatives of the patient, people in charge of monitoring and, of course, the provider of the infrastructure (communication and devices) for the service. This service is composed of a health data transmission service and a videoconference

service while the infrastructure at the patient's home is centered in the Residential Gateway (RGW). A RGW [3] is a small embedded computer running a software platform to manage several services at home, like entertainment applications, video-surveillance, etc. These services are integrated into an OSGi [4] platform that provides also support for remote control and maintenance.

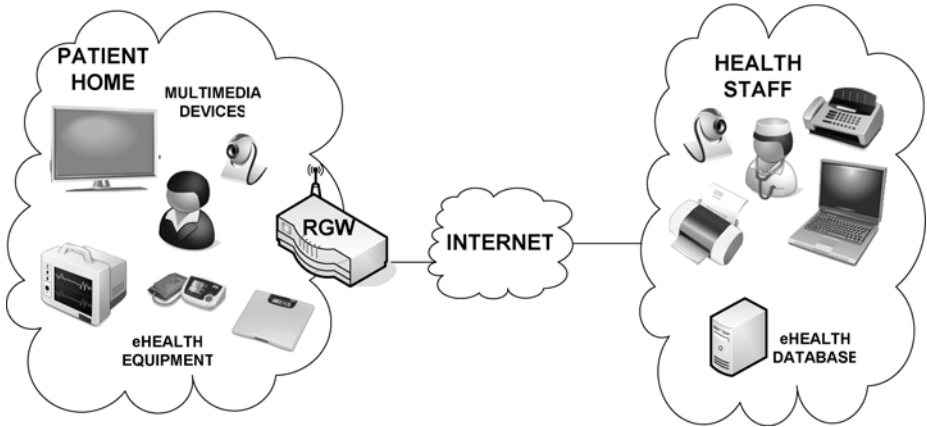


Fig. 1. Overview of the eHealth scenario

In the test scenario implementation, a patient's health data monitoring service is controlled by the RGW and the eHealth equipment is connected via Bluetooth. Any medical information is forwarded to the eHealth Service Provider using HL7 messages. During an on-line medical citation between for example a nurse and the patient, a video call is established. This functionality is based on the UPnP AV standard (Universal Plug & Play for Audio-Video) because it provides a modular framework for multimedia communications and many end-user devices are supporting this standard.

The next sections are organized in the following way: In Section 2, the state of the art of telecare is reviewed. Our proposal is presented in Section 3, describing the platform architecture in more detail. Section 4 presents the application, experiments developed and results obtained. Finally, the last section concludes the paper and gives some future outlook.

2 State of the Art

This section presents the work and research lines related to telemedicine and similar developments followed by a brief introduction to the technologies and standards used in the system.

2.1 Related Research on Telecare and Telemedicine

Home telecare is foreseen as an important factor for future medical assistance [5], [6]. Some telemedicine and telecare approaches are based already on OSGi [7], [8], [9], [10] but they do not offer a complete integration of all services provided in our approach.

For example, the Seguitel system [11] is a social and telecare service platform based on OSGi. It is oriented to provide services designed under a methodology that ensures a SLA (Service Level Agreement) but this approach introduces several middleware layers and it is not covering healthcare standard interoperability. Other projects that work in similar environments like HEALTHMATE [12] (Personal intelligent health mobile systems for Telecare and Teleconsultation), TELECARE [13] (A multi-agent tele-supervision system for elderly care) or PIPS [14] (Personalized Information Platform for Life and Health Services) have similar lacks as Seguitel.

2.2 Technologies

This section provides an overview of the technologies used to achieve the objectives outlined in the introduction. These technologies are mainly the health informatics standards, the OSGi framework, the UPnP Audio Visual standard and the use of Bluetooth for communications with medical devices. They will be briefly described next.

2.2.1 Health Informatics Standards

Home telecare requires that patient data must be transmitted following messaging standard. Currently, HL7 [15], [16] is a widely applied protocol to exchange clinical data. Moreover, there are open source tools available to process and transmit HL7 messages [17], [18].

Furthermore, there is a standard under development, the ISO/IEEE 11073 (also known as x73) standard [19], to transmit medical information among devices, but there are hardly no medical devices yet in the market supporting the standard. Many available devices follow proprietary protocols, so it is not possible to interact with other devices or platforms.

2.2.2 OSGi Framework

The OSGi [4] framework is a Java-based open architecture for network delivery of managed services. Services are added through software components (bundles). The platform carries out a complete management of bundles' life cycle: install, remove, start, stop and update. The bundles are Java applications running on the same JVM (Java Virtual Machine) that can share code.

2.2.3 UPnP AV Standard

The videoconference system allows the communication between the patient and any other member of his group. For example, an assistant or medical personal as well as his relatives can be members of the group. The videoconference functionality needs a multimedia device infrastructure managed by the RGW. The UPnP AV [20] is a standardized UPnP architecture for multimedia systems in home networks. It is a widely spread standard used in multimedia home networks. It allows an automatic discovery of multimedia services with a low CPU usage for a streaming negotiation and management. Additionally, there are open source libraries of the standard available. Other approaches are based on SIP and IMS [21] but UPnP devices are more widely spread in the market.

2.2.4 Bluetooth

The Bluetooth wireless protocol [22] is a short-range communications technology intended to replace wires connecting fixed or mobile devices. The Bluetooth specification supports secure and low power communication for a wide range of devices to connect and transmit information with each other. There are low-cost Bluetooth adapters available in the market as well as medical measurement devices like the UA-767PBT Blood Pressure Monitor from A&D Medical. Thanks to Bluecove, an open-source library that provides a JSR-82 Java interface for Bluetooth Profiles, it is possible to implement OSGi bundles that communicate with Bluetooth devices available for many operating systems.

3 eHealth System Architecture

This section starts with the description of some scenarios covered by the developed system. Finally, an overview of the architecture is presented followed by the presentation of the main elements.

3.1 Telecare Scenarios

Previous telecare proposals are often organized in a way not taking into account the communication with the relatives and friends of a patient. But according to several studies [23], elderly or dependent people are reluctant to use telecare services because they do not personally know the operator or like to contact a person in the telecare service centre. Usability can be increased when incorporating relatives and friends into the flow. In a possible scenario, a doctor initiates a video call with the patient to remotely check some data about the heart health, like the blood-pressure, heart rate or the weight. The RGW keeps an address list of relatives and friends; so, the patient can communicate with them if he needs to contact an emergency service in a serious situation. Moreover, relatives and friends can check medical reminders to help the patient during the treatment.

In summary, the some of the scenarios covered by the telecare service are:

- An elderly man has a medical citation with the doctor to review his heart health.
- An elderly woman that lives alone receives a video call from an assistant or a relative to take care about her.
- The system warns the patient when he has to be prepared for a planned video-conference or when it is time to take a medicine.

3.2 System Overview

Telecare supports the integration of patient-oriented services, like medical data transmission, audio/video calls or healthcare appointment management. Our proposal tries to provide an interoperable and scalable solution. The system is divided in four basic subsystems: Medical, eHealth, Data and Multimedia. These elements are managed by a RGW running with Linux and an OSGi framework hosting different services which can be managed remotely by the telecare or access provider. An architecture schema of the four subsystems is shown in Figure 2.

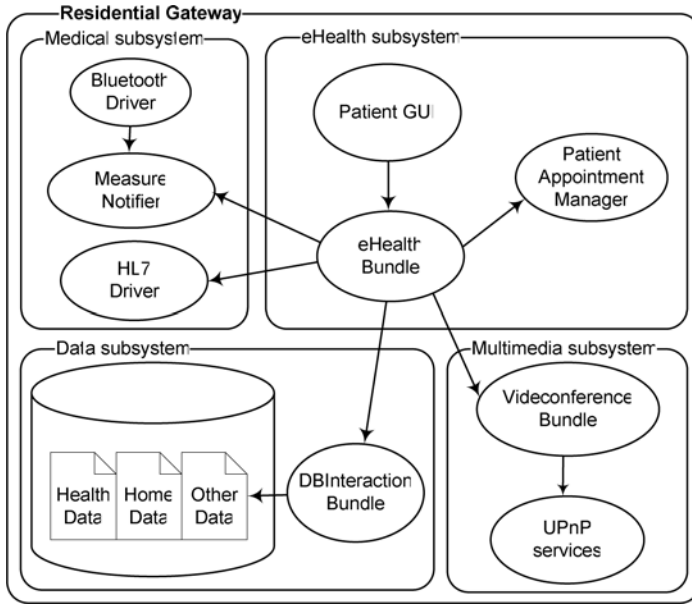


Fig. 2. Architecture schema for the home telecare platform

The Medical subsystem can include a wide variety of devices and protocols. In this approach we have integrated Bluetooth devices to take some measures from the patient. Moreover, it carries out the transmission in HL7 messages. The eHealth subsystem manages the patient’s appointments and medical treatments and implements a graphical user interface (GUI). The Data subsystem includes an SQL Database to save the whole information about patient health data, home sensors data, etc. Finally, the multimedia subsystem establishes the communication between doctors, patients and relatives by means of monitors and webcams.

3.3 eHealth Subsystem

The eHealth subsystem in the patient’s RGW is composed of a set of bundles that includes a graphical user interface, an eHealth bundle and an appointment manager. The Patient GUI is a Swing-based application adapted to the patient and incorporates the eHealth bundle services. This bundle is included also in the doctor and system administrator because these applications share some features. Using the patient’s GUI, is possible to access to a simple patient’s Electronic Health Record (EHR), to look up treatments, medical appointments and remainders.

The patient can launch off-line or on-line medical appointments. In the first case, the patient follows a wizard, for example introducing weight and blood pressure, and waits for the results shown on the display. Health data are sent by the HL7 Driver, so the doctor or nurse can check it later. In an on-line medical appointment, the patient communicates with the doctor or nurse through a video call.

3.4 Medical Subsystem

The Medical subsystem includes a patient healthcare data access module, in our approach; this is a Bluetooth driver, as well as a Measure Notifier and a HL7 driver. The Bluetooth Driver parses the messages sent by the personal scale or blood pressure monitor and the Measure Notifier alerts the eHealth Bundle immediately to take a new measure. This procedure is needed because it could take two minutes for the patient to make the measurement and the Bluetooth Driver receives the data. Measures to be collected, the frequency and their relevance are predetermined by the doctor and the devices available and is reflected in the database of the system.

After a data recovery, the HL7 Driver bundle transmits an ORU-RO1 observation message in HL7 format [24]. The eHealth Server Provider receives and processes the message by a HL7 engine. The HL7 Driver also takes care of sending an ADT-A05 (pre-admit a patient) from the eHealth Server Provider to the RGW when the patient starts to use the telecare system.

3.5 Multimedia Subsystem

In a home telecare scenario, a multimedia infrastructure is required to allow a seamless communication between healthcare actors. As presented in the next section, a multimedia real-time streaming is established between the assistant, the patient and relatives to provide an Audio-Video (AV) call. This infrastructure should be flexible enough to allow several multimedia devices to be connected.

Figure 3 shows the internal design of the Multimedia Subsystem. The AV subsystem handles the AV communication according to the UPnP AV specification. The UPnP Control Point is implemented as an generic UPnP Control Point. The Event

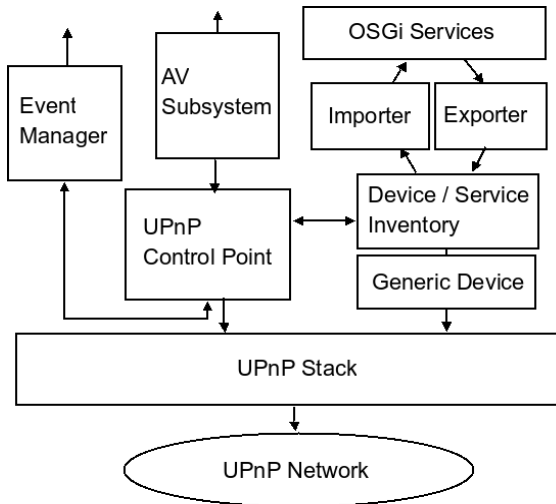


Fig. 3. Multimedia Subsystem Design

Manager handles the events coming from home devices. The Device/Service Inventory is a repository of services that the UPnP Control Point can add and remove. The Importer, imports UPnP services and exports OSGi Service while the Exporter performs the opposite operation: a module that imports OSGi Services and exports UPnP services. Finally, a Generic Device merges and publishes the platform functionality as UPnP services.

Our approach is based in well known multimedia and network standards: UPnP, HTTP (HyperText Transfer Protocol), MPEG-2 and SIP (Session Initiation Protocol). The SIP protocol allows connecting remote devices in dynamic environments like home access networks because the IP address usually is received dynamically and the link status is variable.

4 Experiments

Currently, we have developed and tested a prototype with the functionality described above. First experiments are being implemented in a laboratory with two local networks simulating the communication between two environments, like the patient's home and the ambulance office.



Fig. 4. Graphical interface of the patient application

The simulated RGW is running in an embedded computer with limited resources using an Intel Pentium Celeron CPU, 512 MB memory and Debian Linux. Apache Felix [25] a OSGi R4 Service Platform compliant implementation released under an open source license is chosen as the RGW software platform running over a Java Virtual Machine provided by Sun JDK 1.6. The AV UPnP software is implemented based on a branched version of Cybergarage [26] Java libraries. A simple USB webcam with a microphone incorporated is used to acquire multimedia data. The Medical Information System is simulated by a desktop computer running Mirth Engine application and Apache Felix with the same bundles than in the RGW except the Doctor GUI. HAPI open source libraries are used to implement the HL7 Driver following the HL7 version 2.6 standard. The MySQL database is chosen because its simplicity and robustness; it is managed by the Data subsystem in the RGW and the eHealth Server Provider.

Different graphical applications for the patient and the health professionals have been implemented. They include a basic functionality like citation management, health data transmission and video calls. Figure 4 shows the interface of the application developed for the patient. The first image shows the health data measure wizard for an off-line appointment. The second one shows the patient application during an on-line appointment. To the left there is a simplified Electronic Health Record with a medication list. The video and a contact list are shown in the centre and right hand side. The latest medical appointments are shown at the bottom.

5 Conclusions and Future Work

An eHealth system for a complete home assistance has been presented. Our approach is based on well-known standards, incorporating a personal health data transmission service supporting video calls. It supports health data interoperability also incorporating medical devices located on the patient's site and open to incorporate future devices.

The patient equipment includes a RGW based on open source software and some medical monitor devices. Graphical interface applications for the telecare actors have been implemented including basic functionality as well as a healthcare information system with a medical appointment management and videoconference system to support on-line medical appointments. The videoconference system establishes the communication between patient, doctor and his relatives using cheap end consumer devices.

Future work is directed in two ways. The first one works in the line of increasing the functionality incorporating home sensors and implementing an inference engine to generate medical alerts based on the patient health data. The second line of actuation has relation to the usability of the application. Although the main functionality is working, it is needed to test it with real patients in order to provide a nearest solution to its problems.

Acknowledgments. This work has been partly funded by the Ministry of Industry, Tourism and Trade under the projects Caring Cars FIT-330215-2007-1 (TSI-020400-2008-37), OSAMI Commons ITEA 2 IP07019 (TSI-020400-2008-114), Raudos (TSI-020302-2008-115) and InCare (TSI2006-13390-C02-01).

References

1. European Commission et al.: The Prague Declaration – eHealth 2009 Conference Declaration (2009), <http://www.ehealth2009.cz/Pages/108-Prague-Declaration.html>
2. Norris, A.C.: Essentials of telemedicine and telecare. John Wiley and Sons, Chichester (2002)
3. Hofrichter, K.: The Residential Gateway as service platform. ICCE. In: International Conference on Consumer Electronics (2001) ISBN: 0-7803-6622-0
4. OSGi Alliance, <http://www.osgi.org>
5. Guillen, S., et al.: User satisfaction with home telecare based on broadband communication. *J. Telemed Telecare* 8(2), 81–90 (2002)
6. Biddiss, E., Brownsell, S., Hawley, M.S.: Predicting need for intervention in individuals with congestive heart failure using a home-based telecare system. *J. Telemed Telecare* 15(5), 226–231 (2009)
7. Bobbie, P.O., et al.: Designing an Embedded Electronic-Prescription Application for Home-Based Telemedicine Using OSGi Framework. In: Arabnia, H.R., Yang, L.T. (eds.) *Embedded Systems and Applications*, pp. 16–21. CSREA Press (2003)
8. Clemensen, J., Larsen, S.B., Bardram, J.E.: Developing Pervasive e-Health for Moving Experts from Hospital to Home. In: Proceedings of the IADIS e-Society Conference, Avilla, Spain, pp. 441–448 (2004)
9. Chen, Y., Huang, C.: A Service-Oriented Agent Architecture to Support Telecardiology Services on Demand. *Journal of Medical and Biological Engineering* 25(2) (2005)
10. Wang, F., et al.: Services and Policies for Care At Home. In: *Pervasive Health Conference and Workshops, 2006*, pp. 1–10 (2005)
11. Plaza, P., Sanz, N., Gonzalez, J.: An Optimized eHealth Platform to Provide Electronic Services over Dynamic Networking Environments. In: *Third International Conference on Digital Society (ICDS 2009)*, pp. 1–6 (2009)
12. HEALTHMATE: Personal intelligent health mobile systems for Telecare and Teleconsultation, <http://www.healthmate-project.org/>
13. TELECARE: A multi-agent tele-supervision system for elderly care, <http://www.uninova.pt/~telecare/>
14. PIPS: Personalized Information Platform for Life and Health Services, <http://www.pips.eu.org/>
15. Hutchison, A., et al.: Electronic data interchange for health care. *Communications Magazine, IEEE* 34, 28–34 (1996)
16. Hammond, W.E.: Health Level 7: A protocol for the interchange of healthcare data. In: Moor, G.J.E.D., McDonald, C., Goor, J.N.V. (eds.) *Progress in Standardization in Health Care Informatics*. IOS Press, Amsterdam (1993)
17. HAPI: HL7 application programming interface, <http://hl7api.sourceforge.net>
18. Mirth Corp: Mirth Connect, <http://www.mirthcorp.com/products/mirth-connect>
19. Schmitt, L., Schmitt, L., Falck, T., et al.: Novel ISO/IEEE 11073 Standards for Personal Telehealth Systems Interoperability. In: *Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS-MDPnP 2007)*, pp. 146–148 (2007)
20. UPnP Forum: Universal Plug and Play standard, <http://www.upnp.org>

21. Haber, A., Gerdes, M.: Remote Service Usage Through Sip with Multimedia Access as a Use Case. In: IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007), pp. 1–5 (2007)
22. Bluetooth SIG: Bluetooth, <http://www.bluetooth.com>
23. Hill, S.: Barriers to 'telecare': the perceptions and experiences of workers with responsibility for assessing for, and commissioning, care services and equipment, Report for Essex County Council (2008)
24. De Toledo, P., Lalinde, W., Del Pozo, F., Thurber, D., Jiménez-Fernández, S.: Interoperability of a Mobile Health Care Solution with Electronic Healthcare Record Systems. In: 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, New York, pp. 5214–5217 (2006)
25. Apache Software Foundation: Apache Felix, <http://felix.apache.org>
26. Satoshi, K.: Cybergarage UPnP framework, <http://www.cybergarage.com>

Tracking System Based on Accelerometry for Users with Restricted Physical Activity

L.M. Soria-Morillo¹, Juan Antonio Álvarez-García², Juan Antonio Ortega²,
and Luis González-Abri²

¹ Computer Languages and Systems Dept., University of Seville,
41012, Seville, Spain

² Applied Economics I Dept., University of Seville, 41018,
Seville, Spain

{lsoriamo, jaalvarez, jortega, luisgon}@us.es

Abstract. This article aims to develop a minimally intrusive system of care and monitoring. Furthermore, the goal is to get a cheap, comfortable and, especially, efficient system which controls the physical activity carried out by the user. All this, is based on the data of accelerometry analysis which are obtained through a mobile phone.

Besides this, we will develop a comprehensive system for consulting the activity obtained in order to provide families and care staff an interface through which to observe the condition of the individual subject to monitoring.

1 Introduction

Just 30 minutes of moderate activity five days a week, can improve your health according to the Centers for Disease Control and Prevention¹. By enabling activity monitoring at individual-scale, over extended period of time in a ubiquitous way, physical and psychological health and fitness could be improved. Furthermore, communication among relatives, friends or professionals could be enriched, showing graphics of weekly activity (very interesting for sportsman or elderly's relatives).

Current remote health monitoring applications are in an early commercial stage^{2,3,4} where application programmers, along with medical experts, are trying to analyze diverse parameters for providing wireless automated health care. In such a way, these systems need to transmit data to the backend very often, either by doctors' analysis, or due to computational intensive diagnosis algorithms that can't be executed efficiently on an embedded processor in a wearable device.

Commercial approaches use specific hardware, but we thought that modern mobile phones can achieve the same goals. However, high rates of physiological data have an adverse impact on the phone usability, not only due to expensive long-range communication, but also due to the costly data recovery and battery life.

¹ <http://www.cdc.gov>

² <http://www.fitbit.com>

³ <http://www.directlife.philips.com>

⁴ <http://www.miowatch.com>

With this paper, we hope to develop a dynamic, efficient and reliable system to control the user's monitored activities that have been practiced. Furthermore, the monitoring proposition must be as least intrusive as possible, since users, generally, are adverse to be controlled through traditional surveillance systems like surveillance cameras or sensors for all around their houses. A problem of these systems is the restricted sphere of action.

The system described above, tries to carry out the person's monitoring wherever he goes. This can be carried out thanks to that current mobile devices which incorporate accelerometry sensors, that means the user can carry the device at all times. This does that the range of our application will not be only the subject's house, work or training place, but it could be also controlled wherever he goes.

Our aim is to register every movement practiced by the user and classify it in different activities such as, for example, walking, running, jumping, going up and down stairs or even falls. Once having done that, the result of the classification will be visible by means of a web portal to user's family, doctors and anybody that the system administrator thinks it is necessary.

Furthermore, user's monitoring must carry out without its will be a too heavy load for owner subject. As we will see later, some existing devices allow make an user's monitoring, but the main problem is that only are available through proprietary hardware. Another problem happens as well, the user must wear an additional device; therefore, this can become uncomfortable and increase the risk of being forgotten by the user.

The opposite of above is that our system, on the user's side, expects to be integrated in the user's mobile device monitoring. The advantage of this decision it is just that mobiles are part of user's life style (every day more and more), and thus, the risk regarding loss or oversight is much lower than with an additional device. In addition to this, the boost and versatility of these devices make possible that the system possibilities can be increased.

The mobile devices connectivity is huge nowadays. In addition to the Wi-Fi technology that is more and more integrated in devices, the communication companies are done a strong vouch for 3G connectivity in these terminals. This technology allows us to connect users to Internet wherever they are. Thus, the access to Internet will be available any place and any time. Based on it, we can talk about developing an ubiquitous system that allows us to know subject's diary activity by means of information flowing through Internet.

2 Related Works

The estimated physical activity, according to data obtained which are based on accelerometer, is, at the present, a research topic. There are some systems that are able to carry out this task. Among all of them, we will detail the most important and we will examine it to observe the difference and resemblance in accordance with our proposal.

The first difference we can observe among the developed systems up to now is the situation of the device and the device used for collecting information. There are systems that use proprietary hardware ([1], [2] and [9]), while others use general purpose hardware ([3], [4] and [10]), as the one we are describing. Obviously, to use generic hardware will result in a benefit, because the price and versatility of these devices are tricks to his advantage. However, these devices have a drawback: the limitations on the data collection quality. The accelerometers integrated on generic devices, such as the last generation of devices, have, overall, less quality in the data collection than other accelerometers that are integrated on specific devices. In this way, investigations that use specific hardware must focus on solving this trouble.

Other difference that we can find between the several researches is its own aim. In [5] we can see that the accelerometry sensor is located in a glove, which must wear the user and it is able to recognize several activities based on the hand movement. However, in other researches like [3], the sensor is situated in the user's pocket. Having arrived at this point, we can wonder what the most efficient method is. If we based on results, [5] and [8] has more precision than [3] and, furthermore, it is able to recognize larger number of activities. However, this solution could be uncomfortable for the user, thus the chosen system must be selected depending on the needed required.

In addition to the previous researches, other proposal for people with surveillance exists [7] that is not only based on accelerometry, but it also adds other elements like, for example, surveillance video cameras.

3 System Design

The remote surveillance system will be developed in three different environments: user's terminal, sever and remote query terminal. In the user's terminal will take place the data collection from the accelerometry sensor, data filtered a priori, classification of the user's activity and filtered a posteriori. The server environment will provide query interface for activities that the user is carrying out, communication service with the user's terminal and services needed for the communication with terminal query.

Finally, in query terminal will be executed the information query software. In design of system is contemplated that the software will be of two kinds: web interface or desktop interface. Web interface will be accessible to any user who desire look up the user's activity from any web browser by means of the application server. Desktop interface (both for Smartphone and for PC) will be communicated with the server through the web services offered by the own server.

In the next sections we will explain in more detail in what does consist the monitoring process and the activities control process. But before that, we will show a general view of the system. Figure [1] shows a flowchart of the system framework:



Fig. 1. Diagram of implementation

4 Obtaining and Filtering Data

Following properties are desired for our activity recognizer installed on user device: manageable device, built-in GPS device, accelerometer sensor and its consumption should be enough for recharge the batteries each 24 hours at least.

Advances in microelectronics have reduced the cost of small and accurate sensors. As a result, accelerometers are embedded in modern mobile phones, the most ubiquitous device nowadays. Tri-axial accelerometers can measure G-force every axis and some features could be extracted to recognize the current user's activity.

Other profit of the activity monitoring system based on accelerometry is the low energy consumption produced in the device. This is essential when we talk about developing applications for mobile devices, because the bottleneck of these devices is the batteries. Furthermore, the user does not need recharge it constantly. Thus, the use of the system will be more comfortable for the user.

Specifically, we use the Samsung Omnia for our data collection, which contains a three axis accelerometer with a sensitivity of $\pm 2G$ and a resolution of $0.004g$. Samsung device lasts more than 2 days with continuous use of this accelerometer, so the energy consumption will not be a problem and the user will not be constantly concerned by recharging his/her device.

The fact of having chosen the Omnia device for testing the system is because it integrates all the requirements described earlier: accelerometer, GPS device, operating system and all the facilities to work directly with all the data sensor of accelerometry. Furthermore, it is a current and, relatively, inexpensive device. This ensures us that the system is available for anyone without this meaning, a higher technology spending.

Although the data obtained are completely valid, it will be necessary to carry out a data filtering before making the classification. The filtering aim is to remove all the signal noise. In most of the cases, the noise of the accelerometry sensor is negligible, but exists a kind of noise that can affect seriously to the activity classification. This noise is produced by vibrations that take place on device when it is carried by the user.

We should consider an important aspect of our system: there is no restriction when it comes to choose the place where we must wear the device. This restriction is demanded by any related works. However, in our system, the user can wear the device where he wants. This increases the complexity of the development, because the mobile device subjection couldn't be peak condition. A bad subjection will produce unintended vibrations when the user carries out the activity. This is the reason of executing a filtering, in order to obtain valid data.

Throughout all the movement detection process, it has worked with the module of accelerometry. Usually, most accelerometer tends to measure accelerations that take place at three axis of device (triaxial accelerometer). Figure [2] shows a graphic with the three axis in a generic mobile device.



Fig. 2. Accelerometer 3-axis

Definition 1. Accelerometry vector in time t , denoted as $|a_t|$, is defined as value that will be determined by next equality:

$$|a_t| = \sqrt{a_x^2 \cdot a_y^2 \cdot a_z^2}$$

where a_x, a_y y a_z are values of the acceleration in a certain time instant in the axis x, y, z respectively.

To develop recognition of user activity accelerometry readings have been divided into temporal windows [Figure 3] then, analysis of a time window will result in a possible activity that the user is taking place during the time period that covers the window.

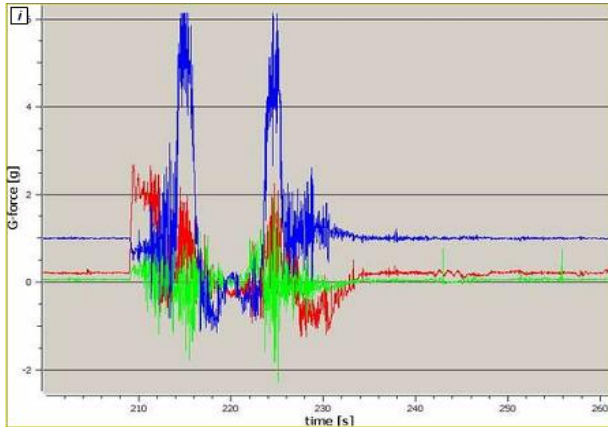


Fig. 3. Accelerometer signal into temporal window

Once recovered a window of 3 seconds of $|a_t|$, Fourier Transform is applied to get the frequency domain signal. Although the size of smaller window is desirable, the low frequency of accelerometer does not allow it (Samsung Omnia accelerometer only has 5Hz).

Cooley-Tukey algorithm was used to reduce the execution time ($O(N \cdot \log N)$). Then, Butterworth low pass filter was applied to eliminate noise caused by device vibrations. Furthermore, the frequency domain signal treatment allows us new readings, using interpolation, a very useful tool when the frequency is as low as our case. The latency of this transformation and filtering is around 2 seconds (in a separate thread) so our recognition activity system has a 5 seconds delay.

5 Classification of the User's Activity

Since the entire process of classification (and training) will be made on the device itself, it is essential to minimize the computational cost. In addition, the design objective recognition of activity patterns is that all logic is done in the terminal. Since the learning until the classification, it should be based on stable pillars and the least heavy possible. This means that the recognition process must be, computationally, the least cost possible and as accurate as possible.

Although there are various techniques that can produce a more accurate of classification, are computationally much more expensive and further learning is, in most cases, unthinkable carry out in a mobile terminal.

The learning and classification method chosen is based on a probability table, and a second level, governed by a dynamic Markov chain which gives more control to the system. Prior to the detection process five vectors formed by six values must be identified.

Definition 2. *The rank vectors of the statistical variable β is defined as one that sets up between its positions a range of values which can group readings values taken from a time window.*

These vectors pretend to define a range for each of the statistical measures presented below:

- Arithmetic mean: $-\infty, 0.8, 0.9, 1.1, 1.2$ and $+\infty$
- Range: $-\infty, 0.24, 0.6, 0.9, 1.2$ and $+\infty$
- Variance: $-\infty, 0.09, 0.15, 0.25, 0.35$ and $+\infty$
- Coefficient of Variation: $-\infty, 10, 25, 38, 45$ and $+\infty$
- Minimum: $-\infty, 0.3, 0.5, 0.7, 0.9$ and $+\infty$

That is, the vector corresponding to the arithmetic mean is composed of the above values that determine five ranges: $[-\infty, 0.8)$, $[0.8, 0.9)$, $[0.9, 1.1)$, $[1.1, 1.2)$, $[1.2, +\infty)$.

In learning, the user must perform each of the activities that can recognize the system. Unlike other works, the number and type of activities are not determined a priori, but the system administrator or users determine the activities that can be recognized. When performing the learning of a particular activity, each vectors corresponding to each of the statistical measure stores the number of temporal windows whose statistical measure is included in that range as well as the activity that is being developed. Thus, after completing the learning process, we have a series of vectors that contain at each position the number of temporal windows of each activity that have been detected in a certain range.

For determinate the activity that the user is carry out, the statistical values described above have been calculated based on the $|a_t|$ content in a given time window. After the expiry of the time corresponding to the time window, we have five values we use to determine the statistical activity that has taken place by the user during that time period. The next step is to visit the array of frequencies which are generated during the learning process. There will be a matrix for each activity and shall consist of N tables (where N is the number of activities that the system can recognize) composed by five rows (one for each statistical measure) and five columns (one for each range defined by the ranks vectors).

Definition 3. *The matrix of frequencies associated with ∂ activity is defined as that matrix which reflected in the position $[i, j]$ the number of readings that have been collected in the learning process of the activity ∂ of the statistical variable i in range j . J As the range defined in the vectors of range. After each process of learning, the values that made up the matrix are normalized to avoid dependence on the training time of each activity.*

Having analyzed the current time window, we resort to the matrix learning ∂ of each activity. We will make a summation of those positions in the matrix in which the range of values of the statistical variable of row i , coincides with the value obtained from the time window. With this we get a number denoted as $\Omega_t(\partial)$.

Definition 4. *The Contest Sum, denoted by $\Omega_t(\partial)$, is defined as the contents sum of the matrix learning activity ∂ positions where the value obtained in the analysis of the*

time window t for the statistical measurement i coincides with the range defined in the standard position j .

Therefore, at this point, the probability of such activity is known based on reading conducted by accelerometry sensor and the study of the time window generated. It is also important bear in mind the reduced runtime of algorithm of detection. The algorithm used has linear complexity, so that its mobile terminal development is possible and would not entail excessive computational cost.

Definition 5. Denoted by $\ddot{\delta}$ the most likely based activity $\Omega_t(\delta)$ obtained for each activity $\hat{\delta}$:

$$\ddot{\delta} = \max_{\hat{\delta}} \Omega_t(\hat{\delta})$$

To clarify the process of classification of the user’s activity, an example will be made below. First, the system collects accelerometry data every 200 milliseconds, although this frequency can be adjusted on the system configuration to allow lower frequency reading if the device support it. Every time the system get a data, its magnitude is calculated as it was detailed in *Definition 1*. This module will be stored in an array created for that purpose. This will happen during the time set for the time window, so after this period the vector of modules will be filled by all the readings taken. Then, from the stored data, statistical values listed in *Definition 2* are calculated, i.e., *Arithmetic Mean, Range, Variance, Coefficient of Variation and Minimum*. For the present example, suppose that the values for these fields are:

- *Arithmetic Mean:* 0.87
- *Range:* 0.7
- *Variance:* 0.21
- *Coefficient of Variation:* 35
- *Minimum:* 0.2

Now it's time to compare these values with the matrix described in *Definition 3*. To continue the previous example, assume that the system is able to recognize three activities: running, walking and jumping. In this way, our cube matrix shall consist of three two-dimensional arrays (one for each activity can recognize) 5x5 size (number of intervals for each statistical variable calculated and number of statistical variables respectively). Three-dimensional matrices resulting from the completion of the learning process are shown below:

| Running | Interval [1] | Interval [2] | Interval [3] | Interval [4] | Interval [5] |
|--------------------------|--------------|--------------|--------------|--------------|--------------|
| Arithmetic mean | 0 | 21 | 14 | 29 | 16 |
| Range | 0 | 0 | 17 | 30 | 33 |
| Variance | 0 | 10 | 20 | 35 | 15 |
| Coefficient of Variation | 20 | 36 | 10 | 14 | 0 |
| Minimum | 5 | 27 | 28 | 20 | 0 |

| Walking | Interval [1] | Interval [2] | Interval [3] | Interval [4] | Interval [5] |
|--------------------------|--------------|--------------|--------------|--------------|--------------|
| Arithmetic mean | 10 | 35 | 20 | 15 | 0 |
| Range | 10 | 15 | 27 | 18 | 10 |
| Variance | 7 | 18 | 31 | 14 | 10 |
| Coefficient of Variation | 5 | 10 | 37 | 19 | 9 |
| Minimum | 31 | 20 | 15 | 8 | 6 |

| Jumping | Interval [1] | Interval [2] | Interval [3] | Interval [4] | Interval [5] |
|--------------------------|--------------|--------------|--------------|--------------|--------------|
| Arithmetic mean | 15 | 19 | 27 | 13 | 6 |
| Range | 18 | 21 | 30 | 11 | 0 |
| Variance | 10 | 16 | 26 | 15 | 13 |
| Coefficient of Variation | 3 | 24 | 27 | 18 | 7 |
| Minimum | 29 | 30 | 13 | 8 | 0 |

Once presented the learning matrix, based on *Definition 4*, the sum of learning cases for each of the activities that the system is able to recognize based on data obtained after analyzing the current time window are obtained. In the previous matrix, values in bold correspond to those intervals in which the group of values of the time window analyzed are contained. Based on the values obtained, the sums $\Omega_t(\partial)$ of each of the following activities will be:

- Running: $\Omega_t(Run) = 21+17+20+10+5 = 73$
- Walking: $\Omega_t(Walk) = 35+27+31+37+31 = \mathbf{161}$
- Jumping: $\Omega_t(Jump) = 19+30+26+27+29 = 131$

Once obtained the $\Omega_t(\partial)$ for each of the activities, based on *Definition 5*, most likely activity ($\check{\partial}$) is chosen, whose sum $\Omega_t(\partial)$ has maximum value. For the present example, this activity is *walking*.

6 Classify Filter

Most probable activity was recognized using accelerometer data, but the result of this classification suffers from noise and transitions between activities generate abrupt oscillations. A Markov chain will be introduced to control in a higher level the final classification of most probable activity.

The selection of a Markov chain to model the user’s activity is based on the fact that if a user is doing an activity, there will be other related activities that he/she will do with a higher probability. For example, when a user is *sitting*, still *sitting* will be more probable than change to *jumping* activity.

Markov chain states will be the different activities that system could recognize. Given state ∂_i , probability of transition to state ∂_j will determine transition probabilities of Markov chain (p_{ij}). That is to say, change probability from one activity to other or the same every 3 seconds due to duration time window.

Obviously assigned probabilities to transition matrix are static, that is to say, they are defined by the system before beginning the activity recognition. However, *dynamic probability index* is proposed in this work:

Definition 6. *Dynamic probability index for transition between ∂_i and ∂_j will be denoted by α_{ij} and will be defined as the hope that activity ∂_j is done on current state ($\Omega_t(\partial_j)$) multiplied by the transition probability between ∂_i and ∂_j and normalized:*

$$\alpha_{ij} = \frac{\Omega_t(\partial_j) \cdot p_{ij}}{\sum_{j=1}^n (\Omega_t(\partial_j) \cdot p_{ij})}$$

From this moment, when we talk about transition probability of Markov chain from ∂_i to ∂_j , we will refer to α_{ij} .

Definition 7. *Markov chain's transition matrix then will be denoted as T :*

$$T = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1j} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2j} \\ \vdots & \vdots & \dots & \vdots \\ \alpha_{i1} & \alpha_{i2} & \dots & \alpha_{ij} \end{bmatrix}$$

We will define the **state vector** in instant t and we will denote as α^t to the vector of length j , where j is the number of activities that system can recognize and where each position of the vector corresponds to the probability that a transition to activity j (∂_j) in instant t is done from activity done in instant $t-1$:

$$\alpha^t = [\alpha_j^t] = \left[\sum_{i=1}^n \alpha_j^{t-1} \cdot \alpha_{ij} \right] = \alpha^{t-1} * T$$

Once defined the state vector, we can select the most probable activity.

Definition 8. *We will define it as the current recognized activity and we denoted by $\hat{\partial}$ to the activity ∂ for which component α^t is larger:*

$$\hat{\partial} = \max_j \alpha_j^t$$

It will be noted that the recognized activity after all the classification process is $\hat{\partial}$, and it must not to match to ∂ , due to the defined Markov chain includes additional information to the model.

To summarize, a very efficient Naïve classifier made an initial classification to then pass the output to a Markov chain that eliminates noise based on temporal knowledge of previous activity and the likelihood of transitioning into next activity.

7 Server

Once the classification of the activity has been completed and the system has recognized the activity, that the user is doing, it is time to send this information to the server. The aim is that the information was available to all users that have permission to look it. Thus, to control the activity of the user is possible.

This requires first to describe the architecture of the server in order to know the interactions available to it. The server will have a dual function. On one side will have a web services interface with which the system will be able to communicate with the user's mobile device monitoring, so that it is possible to send information about the activity which is being carried out and the time it began. On the other hand offer a web interface and a set of web services through which other users could view the information of activity that user is carrying out, either through a web browser or through special software that makes use of those services. Figure [4] shows the structure of the server.

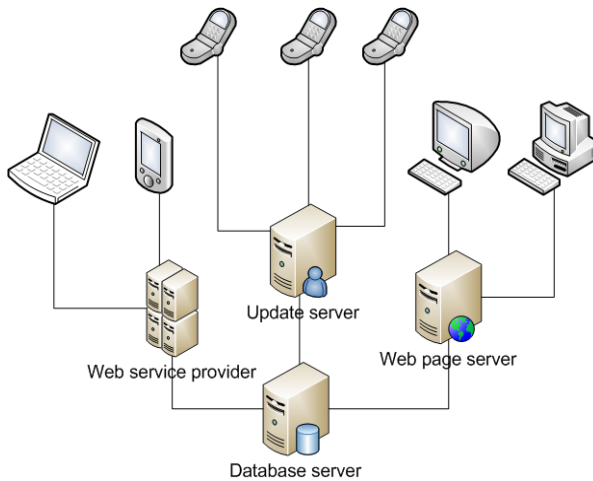


Fig. 4. Server diagram

The updated server will contain a set of web services that will allow the application, for recognizing the user's activity, can send the activity information to the server and store it in the database. For them will be defined at least one function: Updating activity. The parameters of this function are the time and date on which the user began the activity and the identifier of the activity itself with the ID of the user who performed it. This feature adds to the database a new entry with the data provided.

We must make an explanation before continue. The continuous transmission of information about the activity carried out could make the battery of the user mobile user expires quickly. Therefore, only new user activity will be sent on the servers when it change. That is, if an user left the device on the bedside table, for example, the last activity would be sent "without device", which would occur by detecting transmission that has left the device on a surface. Subsequently, it does not send any information to

the server for the rest of the night, until the user returns to pick up his/her device to start the daily activity. In case that the device loses the connection with the server for any reason, for example the failure of network access, the activity performed by the user will be stored in the mobile device next to the time at which such activity occurred. As soon as the connection to the server is refreshed, such activities will be uploaded to the server to update the record of user's activities. With this process we increase the reliability of the system for consultancy work, as no false readings assigned activities due to failed connections to the server. First, the use of a keep alive signal was issued, but the idea was rejected in order to avoid overloading the server. Also, keeping alive signal would be meaningless if the above technique was used, because the data would be updated automatically once the server connection is restored.

This web service makes it possible to update the user's status, but not consulted. To this end, the web service provider and Web page server was developed. Both are connected to the database and obtain from it, the necessary information. Normally this information is the activity undertaken by a particular user during a certain time. In addition, it provides the ability to generate statistics about activities, duration and level of physical activity based on user activities detected.

8 Experimental Results

Results about evaluation of the system are presented below. On one side will discuss the effectiveness and efficiency of the proposed architecture and secondly, the reliability and accuracy of the recognition of physical activity system proposed.

The activity of recognition based on the *Naive-Bayes* classifier with the noise reduction system based on the *Cooley-Tukey* and *Butterworth* filter has produced excellent results even with the presence of high noise levels.

To check the accuracy of the method, the system has been trained to be able to recognize 5 activities of daily life of any person: walking, running, jumping, going up and down stairs. The learning undertaken by all users subject under study has consisted of 15 minutes per activity, excluding jumping activity for which training has only been 5 minutes.

The following table lists the results of classification based on the number of temporal windows correctly and incorrectly recognized in the detection process:

Table 1. Activity recognition results

| Activity | Success | Failure | % success |
|--------------|---------|---------|-----------|
| Walking | 552 | 37 | 94.00% |
| Running | 258 | 18 | 93.00% |
| Stopped | 478 | 4 | 99.00% |
| Climb stairs | 331 | 25 | 93.00% |
| Down stairs | 441 | 41 | 92.00% |

The results show that the prototype for detection of physical activity has got some very good results despite a short training. However, opposite to what one might think, a long learning can negatively affect the outcome of the classification, due to *over-learning*. This effect usually occurs when the number of training data is very high for certain activities while others too low. This problem causes that the less trained activities are more concrete when *Naive-Bayes* method described above makes the classification, while more trained activities have a higher probability in all ranks of classification determined by the *rank vectors*. Figure 5 shows a comparison between our systems under two conditions: normal learning (with similar training times for each activity) and overlearning with walking activity much more trained. On the x-axis we can see the activities that the user has done during one hour. On the y-axis are shown the number of activities of each type recognized, this axis leads us to determinate the accuracy of recognition system. Finally, each color represents the activity recognized on an overlearned system, on a normal system and the true activity that user was doing.

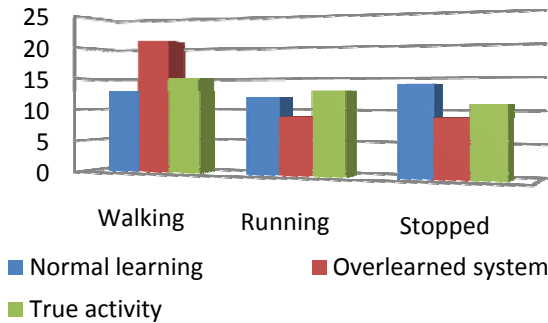


Fig. 5. Overlearned system

9 Conclusions

We have achieved to perform a comprehensive system able to recognize, classify and share information about physical activity in a group of subjects. We have thus succeeded in controlling the activity of a user with a restriction of movement. This is very common in athletes who are in a period of restricted activity (e.g. rest) or elderly, since, thanks to the system, we can control anytime their activity. In other words, the activity that users are doing can be viewed by care persons without the need of being there, at the same place as user, allowing later the analysis of information of their activities.

Moreover, the entire system has been adapted to a goal: to be installed on mobile devices. This requirement has led to all methods of detection, classification and recognition of activities which must have a reduced computational cost. Due to the above, not only the risk of saturation of the mobile device system due to excessive calculations is reduced, but also power consumption is reduced to a minimum. This makes the user must recharge the device less frequently and therefore the system will be more functional and comfortable.

Finally, the proposed designed system has provided the capacity to access to the information of assisted subject through several ways. In this way, multitude of different devices (phones, PCs, PDAs, etc.) are covered by the system in order to access to the information anywhere and anytime.

Acknowledgement

This research is partially supported by the MCI I+D projects FAMENET InCare (TSI2006-13390-C02-02) and ARTEMISA (TIN2009-14378-C02-01) and Andalusian Excellence I+D project CUBICO (TIC2141).

References

- [1] Paiyaram, S., et al.: Activity Monitoring System using Dynamic Time Warping for the Elderly and Disabled people. In: 2nd International Conference on Computer, Control and Communication (February 2009), ISBN: 978-1-60558-792-9
- [2] Ravi, N., et al.: Activity Recognition from Accelerometer Data. In: American Association for Artificial Intelligence (2005), ISBN: 1-57735-236-x
- [3] Hong, Y.-J., et al.: Activity Recognition using Wearable Sensors for Elder Care. In: IEEE Future Generation Communication and Networking (2008), ISBN: 978-0-7695-3431-2
- [4] Yang, J.-Y., Wang, J.-S., Chen, Y.-P.: Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. In: Pattern Recognition Letters (August 2008), ISSN: 0167-8655
- [5] Brezmes, T., Gorricho, J.-L., Cotrina, J.: Activity Recognition from Accelerometer Data on a Mobile Phone, June 2009. LNCS. Springer, Heidelberg (2009), ISBN: 978-3-642-02480-1
- [6] Hyun, J., et al.: Estimation of Activity Energy Expenditure: Accelerometer Approach. In: IEEE: Engineering in Medicine and Biology 27th Annual Conference (September 2005)
- [7] Cho, Y., et al.: SmartBuckle: Human Activity Recognition using a 3-axis Accelerometer and a Wearable Camera. In: HealthNet 2008: Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments, pp. 1–3 (2008), ISBN: 978-1-60558-199-6
- [8] Cho, I.-Y., et al.: Development of a Single 3-Axis Accelerometer Sensor Based Wearable Gesture Recognition Band. LNCS. Springer, Berlin (ISBN: 978-3-540-73548-9)
- [9] Olsen, G., Brilliant, S., Primeaux, D., Najarian, K.: Signal processing and machine learning for real-time classification of ergonomic posture with unobtrusive on-body sensors. In: ICME International Conference on Complex Medical Engineering (CME 2009), April 9-11, pp. 1–11 (2009)
- [10] Joo Hyun Hong, N.J.K., Cha, E.J., Lee, T.S.: Classification Technique of Human Motion Context based on Wireless Sensor Network. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2005)

Web Query Reformulation Using *Differential Evolution*

Prabhat K. Mahanti¹, Mohammad Al-Fayoumi², Soumya Banerjee³,
and Feras Al-Obeidat¹

¹ Department of CSAS, University of New Brunswick, New Brunswick, Canada

² King Abdul Aziz University, Faculty of Computing & Information Technology, Saudi Arabia

³ Department of Information Technology, Birla Institute of Technology Mesra, India

Abstract. This paper presents a query reformulation and clustering technique using *Differential Evolution*. *Differential evolution (DE)* has emerged as one of the fast, robust, and efficient global search heuristics of current interest. The proposed *DE* automatically determines the type of a query and new pattern of query reformulation.

Keywords: Query Reformulation, Differential Evolution, Clustering.

1 Introduction

Information retrieval is an interactive and iterative process and single query seldom satisfy users query on web. In that case, the user has to *reformulate* his/her initial query because it was over or under-specified, or did not use terminology matching relevant documents, or simply contained errors or typos. The query information can be analyzed from query log and better search experience can be constructed through search engine. It has been observed, whenever the user enters two queries in sequence; the connection or relation between the two repetitive queries can be defined as a *query transition*. If the user retains in the same search paradigm, then this connection is referred as a *query reformulation*. The goal of this work is to create a query reformulation model using evolutionary computational technique, and thus the first task is to define the target categories of the proposed model. Conceptually, if the two sequential queries of users demonstrate broader syntactic and semantic gap, then the proposed model performs a query slice clustering on query flow graph technique using *Differential Evolution algorithm*. Most of the existing clustering techniques, based on evolutionary algorithms, accept the number of classes K as an input instead of determining the same on the run. Nevertheless, in many practical situations, the appropriate number of groups in a previously unhandled data set may be unknown or impossible to determine. For example, it is found that, while clustering a set of retrieved documents arising from the query to a search engine, the number of classes K changes for each set of each document. In addition to, if the data set is described by high-dimensional feature vectors like as web centric data, it may be practically impossible to visualize the data for tracking its number of clusters. The context of the present query reformulation envisages into the query graph. The information extracted from query logs can be summarized and suitably represented through query graphs. The query graph is bipartite, with nodes representing queries and documents, and with

an arc connecting a query q and a document d if and only if d was clicked by some user after submitting the query q . This describes an application of *Differential Evolution* to the automatic segmentation and clustering of large unlabeled web data sets to identify the pattern of completeness of query. In contrast to, most of the existing clustering or segmentation techniques, the proposed algorithm requires no prior knowledge of the data to be segmented. Rather, it determines the optimal number of partitions of the data on the “fly”. We apply the proposed model to a large query logs to investigate the new patterns and similar segment relationship during the reformulation of query by the user. The validation of the algorithm is able to produce more pattern of query reformulation and thus improve the better experience of searching. The rest of the paper has been organized as follows: Section 2 elaborates the problem with examples. The related works have been discussed in section 2.1. Section 3 describes mathematical background of the proposed *Differential Evolution* technique. Section 4 presents proposed algorithm and section 4.1 validates the proposal with a public data set of query log. Finally, section 5 gives conclusion and further scope of research.

2 Statement of Problem

The query reformulation can be represented as zero dissimilarity (same query with the induction of error, like the user switches to capital letter or spelling has been modified (example: “indai” and “India”). There is second possibility of query reformulation, where the wordings of query have been altered, but keeping exactly the same target in the sense of presentation of query. Example includes “*Cheap Hotel*” and “*Hotel with Minimum Cost*”. Even there are unconditional changes in query presentation in sequence like “*Tourist Spots of India*”, “*Mobil Connections providers in India*”. Finally, the complete query could be modified by the user both in syntax and semantics. The basic stages of query reformulation are shown in Fig. 1.

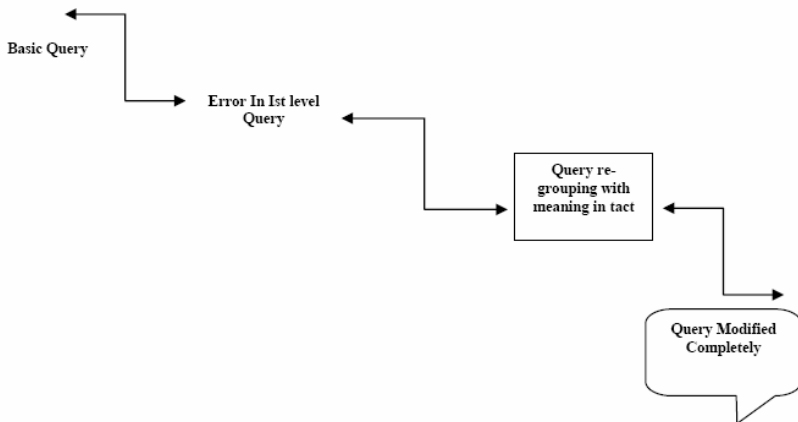


Fig. 1. Query Reformulation Types

The variation in query string also may differ in the new query formation with added generalization level. That means, the modified query will be more general to understand the search pattern of user. For an example the query on “*Mountaineering*”, can be made more generalized to query “*explore Mountains*”. In reverse direction, the specialization also can augment query reformulation of user, like “*Catalogue of Furniture*”, to “*iron grid furniture*”. The generalization and specialization both can be expected from the user’s generalization and ambition to recall and reformulate the query, whereas a specialization is the need to improve precision and reduces the search space. Practically they form a transitive relation and that must be anti-symmetric in nature. To analyze all hidden and latent variation, a sound clustering and query segmentation technique is required prior to parse the final query through search engine. Therefore, the proposed model concentrates on query logs and infers the hidden semantics of user interactions with search engines.

2.1 Related Works

Baeza- Yates [1] identifies five different types of graphs. In all cases, the nodes are queries; a link is introduced between two nodes respectively if:

- the queries contain the same word(s) (*word graph*),
- the queries belong to the same session (*session graph*),
- users clicked on the same URLs in the list of their results (URL¹ *cover graph*),
- there is a link between the two clicked URLs (*URL link graph*)
- There are l common terms in the content of the two (*link graph*).

These graphs can be effectively utilized for segmenting and clustering the session based on the user’s search pattern. The study of query reformulation types started with the work of Lau and Horvitz [4], who sampled 4960 queries from a query log and manually labeled the transitions they found, proposing a classification of query-reformulation types. Rieh and Xie [5] manually labeled 313 search missions and suggested a more fine-grained classification. The information extracted from query logs can be summarized and suitably represented through query graphs, several examples of which are cited in [6][7]. Recently, Boldi et al elaborated the concept of query reformulation and suggested learning phenomena in the pattern itself [8]. To improve the analysis of accuracy and effective clustering of query reformulation, the present work introduces the evolutionary technique in the form of *Differential Evolution*.

The next section describes the essential backend components, which are deployed in the proposed model.

2.2 Essential Components of Proposed Model

For the proposed model, we define following components mathematically:

Query Log:

A query log records information about the search actions of the users of a search engine. Such information includes the queries submitted by the users; documents are

¹ URL: Unique Resource Locator

viewed as a result to each query, and documents clicked by the users. A typical query log L is a set of records $\langle q_i, u_i, t_i, V_i, C_i \rangle$, where: q_i is the submitted query, u_i is an anonymized identifier for the user who submitted the query, t_i is a timestamp, V_i is the set of documents returned as results to the query, and C_i is the set of documents clicked by the user [2].

- **Sessions:** A user query session is defined as the sequence of queries of one particular user within a specific time limit. More formally, if t_0 is a timeout threshold, a user query session S is a maximal ordered sequence: $S = \langle q_{i1}, u_{i1}, t_{i1} \rangle, \dots, \langle q_{ik}, u_{ik}, t_{ik} \rangle$, where $u_{i1} = \dots = u_{ik} = u \in U$, $t_{i1} \leq \dots \leq t_{ik}$, and $t_{ij+1} - t_{ij} \leq t_0$, for all $j = 1, 2, \dots, k - 1$.
- **Chains:** A chain is a topically coherent sequence of queries of one user. Radlinski and Joachims [3] defined a chain as “a sequence of queries with a similar information need”. Unlike the concept of session, chains involve relating queries based on the user information need, which is an extremely hard problem.
- **Pattern:** A pattern is a physical or abstract structure of objects. It is distinguished from others by a collective set of attributes called features, which together represent a pattern [9]. Let $P = \{P_1, P_2, \dots, P_n\}$ be a set of n patterns or data points, each having d features. These patterns can also be represented by a profile data matrix $X_{n \times d}$ with nd -dimensional row vectors.
- **Query Recommendation after Reformulation**
A simple recommendation scheme that uses the query flow graph is to pick, for an input query q , the node having the largest $w_0(q, q_0)$. The query-flow graph G_{qf} is a directed graph $G_{qf} = (V, E, w)$ where: the set of nodes is $V = Q \cup \{s, t\}$, i.e., the distinct set of queries Q submitted to the search engine and two special nodes s and t , representing a starting state and a terminal state which can be seen as the begin and the end of a chain.

3 Differential Evolution Technique: State of the Art

In this paper, the concept of *Differential Evolution (DE)* has been used to determine the automatic clustering in different query stages and finally the query graph is formulated with more in depth pattern behavior of search. The classical *DE* is a population-based global optimization algorithm that uses a floating-point (real-coded) representation.

The i^{th} individual vector of the population at time-step (generation) t has d components (dimensions), i.e., $\vec{Z}_i(t) = [Z_{i,1}(t), Z_{i,2}(t), \dots, Z_{i,d}(t)]$. For each individual vector $\vec{Z}_k(t)$ that belongs to the current population, DE randomly samples three other individuals, i.e. $\vec{Z}_i(t)$, $\vec{Z}_j(t)$ and $\vec{Z}_m(t)$, from the same generation (for distinct k, i, j , and m). It then calculates the (component wise) difference of $\vec{Z}_i(t)$ and $\vec{Z}_j(t)$,

scales it by a scalar F (usually $\in [0, 1]$), and creates a trial offspring $\bar{u}_i(t+1)$ by adding the result to $\bar{Z}_m(t)$. Thus, for the n th component of each vector:

$$U_{k,n}(t+1) = \left\{ \begin{array}{l} Z_{m,n}(t) + F(Z_{i,n}(t) - Z_{j,n}(t)), \text{ if } \text{rand}_n(0, 1) < Cr \\ Z_{k,n}(t), \text{ otherwise.} \end{array} \right. \quad (1)$$

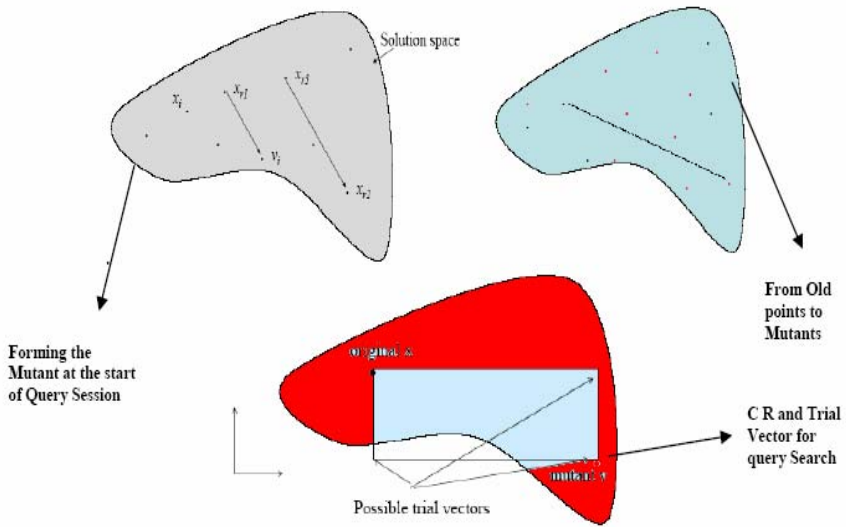


Fig. 2. Differential Search space in Query processing (3 Cases)

$Cr \in [0, 1]$ is a scalar parameter of the algorithm, called the *crossover rate*. If the new offspring yields a better value of the objective function, it replaces its parent in the next generation; otherwise, the parent is retained in the population. For n data points, each d dimensional, and for a user-specified maximum number of clusters set in query reformulation K_{max} , a chromosome is a vector of real numbers of dimension $K_{max} + K_{max} \times d$. The first K_{max} entries are positive floating point numbers in $[0, 1]$, each of which controls, whether the corresponding cluster is to be activated (i.e., to be really used for classifying the query type for each session) or not. The remaining entries are reserved for K_{max} cluster centers, each d dimensional. Thus, we select an optimal threshold in query cluster; in the form of Threshold_{ij} forms the following prototype rule:

IF $\text{Threshold}_{ij} > 0.5$, **THEN** the j^{th} cluster center (of query at session q) \bar{m}_{ij} **is ACTIVE,**
ELSE is INACTIVE **(Rule 1)**

3.1 Modeling the Query Log Components with *DE*

In order to apply *DE* in our proposed model, we consider the following initial assumptions on web query encountered both in session related and temporal features; and add the textual assumptions:

- Number of sessions in which reformulation (q, q') occurs;
- divided by the number of sessions in which (q, x) occurs (for any x);
- among all sessions containing (q, q');
- average number of clicks since session begin, and since the query preceding (q, q');
- average session size of other sessions containing (q, q'); average position in session expressed as number of queries before q since the session begun.

The assumptions also depict that the initial error in query is reformulated to grouping construct and finally the query is either generalized or specialized and may lead to complete final modifications. Hence, the temporal features could also be considered like:

- average time elapsed between q and q' in each session in which both occur;
- Sum of $1/t_i$ where t_i is the elapsed time between queries i and the previous event in a session.

Considering these assumptions the pseudo code for complete proposed model is given:

Step 1. Initialize each chromosome to contain K number of randomly selected cluster centers of query string at the 1st level and K (randomly chosen) activation thresholds in [0, 1] in session q.

Step 2. As the 1st reformulation of steps the grouping is done (in 2nd reformulated session q') keeping the semantics in tact and to find out the active cluster centers off the string (e.g. "Cheap Hotel" and "Hotel with Minimum Cost") in each chromosome of the *DE* with the help of the Rule 1:

Step 3. for $t = 1$ to t_0 do /* t_0 is a timeout threshold, a user query session S*/

for each query vector \bar{Q}_v , calculate its distance metric $d(\bar{Q}_v, \bar{m}_{ij})$ from all active query cluster centers of the i^{th} chromosome \bar{V}_i

/* According to Equation 1*/

end for ;

end for ;

Assign \bar{Q}_v to that particular cluster center \bar{m}_{ij} ,

$$\text{Where, } d(\bar{Q}_v, \bar{m}_{ij}) = \min_{\forall b \in \{1, 2, \dots, k\}} d(\bar{Q}_v, \bar{m}_{ij})$$

if the number of data points that belong to any cluster center $m_{i,j} > 2.5$.

/* Heuristic Test Value of Threshold*/

Update the cluster centers of the chromosome using the concept of average;

/*n/K data points*/

Apply Candidate Solution Measure

/ According to Equation 2 */*

end if

The proposed model has been implemented in *Python Script* (OpenEye-python-1.4.2-1-microsoft-win32-msvc-i686, on PIV 2.2-GHz PC, with a 512-KB cache and a 2-GB main memory in Windows Server 2003 environment. The validation and performance appraisal of the algorithm is discussed in next section.

4 Test Dataset and Validation of Results

The example data set of typical query classification using *DE* has been prepared. The categorization is done on the basis of the query in session 1 and reformulated query in session 2 (Refer Table 1). After parsing to the proposed *DE* classifier, the type of query could be intermediately predicted. It should be noted that in this data set, the reformation for typographic errors has not been included for classification.

Table 1. Query Session Classification Using *DE* Example Data Set

| Search Session 1 q | Reformulated Search Session 2 q' | Query to DE classifier - Generalized or Specialized |
|-------------------------------------|---------------------------------------|-----------------------------------------------------|
| Indian Paneer Hill resorts In India | Chinese Tofu North India Hill Resorts | Generalized Specialized |
| National Bird | National Geo graphic Channel | Generalized |
| Exported Items | Exported Food Items | Specialized |

The parameter set for proposed *DE* is shown in Table 2. All the essential parameters like population size, cross over rate, K_{max} and K_{min} cluster number for query and reformulation of query has been fixed from the implementation aspects of *DE*.

Table 2. Optimal Parameter Setting for the Proposed Model

| Parameter | Value |
|-----------------------------------------------------|--------|
| Population Size | 10*Dim |
| Cross Over Rate on Query Transition | 0.9 |
| Scale Factor on Query reformulation | 0.8 |
| user-specified maximum number of clusters K_{max} | 20 |
| user-specified maximum number of clusters K_{min} | 2 |

The final predicted clusters for reformulated cross over queries for two consecutive sessions can be calculated through Candidate *Solution* measure. Chou *et al.* have proposed the *Candidate Solution* (CS) measure [10] for evaluating the validity of a clustering scheme. Before applying the CS measure, the centroid of a cluster is computed by averaging the data vectors that belong to that cluster using:

$$\vec{m}_i = \frac{1}{N_i} \sum_{x_j \in C_i} \vec{x}_j.$$

A distance metric between any two data points \vec{X}_i and \vec{X}_j is denoted by $d(\vec{X}_i, \vec{X}_j)$. Then, the CS measure can be defined as:

$$CS(K) = \frac{\frac{1}{K} \sum_{i=1}^K \left[\frac{1}{N_i} \sum_{\vec{X}_t \in C_t} \max_{\vec{X}_q \in C_t} \{d(\vec{X}_i, \vec{X}_q)\} \right]}{\frac{1}{K} \sum_{i=1}^K \left[\min_{j \in K, j \neq i} \{d(\vec{m}_i, \vec{m}_j)\} \right]} \tag{2}$$

After applying the Candidate Solution measure, we implement the proposed algorithm of DE on “spring 2006 Data Asset” distributed by Microsoft Research, we built the query-flow graph from this dataset (script shown in appendix to extract query log). A set of 120 input queries were selected. Out of which, some complex queries in both sessions are presented in Table 1.

Table 3. Mean and Standard deviation over 50 independent runs assuming at least 2 query session s for Reformulation

| Dataset | Algorithm | Av. No. of Query reformulation Cluster found | CS measure | Mean intra Cluster Distance | Mean inter Cluster Distance |
|---------------------------------------------|---------------------------------|----------------------------------------------|--------------------|-----------------------------|-----------------------------|
| spring 2006 Data Asset 120 input queries | Proposed Differential Evolution | 3.44 ± 0.0139 | 1.7132 ± 0.0772 | 4.6543 ± 1.31312 | 2.6221 ± 1.356 |

From practical point of view, it is suggested to set the number of parents to 10 times the number of parameters, select weighing factor: F=0.8 and cross over constant CR=0.9.

From the plot given in Fig.3, it is shown that different scale factor (F) associated with query and reformulation of query for session q and q' for entire generation of DE. The convergence of query seems varying for each different value of F starting

from 0.2 to 1.5. Here, we deliberately set $CR = 0.3$ to demonstrate that the variation of the value of cross over rate from standard 0.9 to 0.3 will affect the segmentation and classification of query cluster reformulation strategy. This has also been inferred from the empirical result of query reformulation and convergence that increase in number of population and simultaneously lower the scale factor; convergence is more likely in proposed *DE* (for each query cluster per session to occur) but generally it takes longer time. Thus, the algorithm shows a trade-off between robustness and speed of convergence in terms query reformulation strategy.

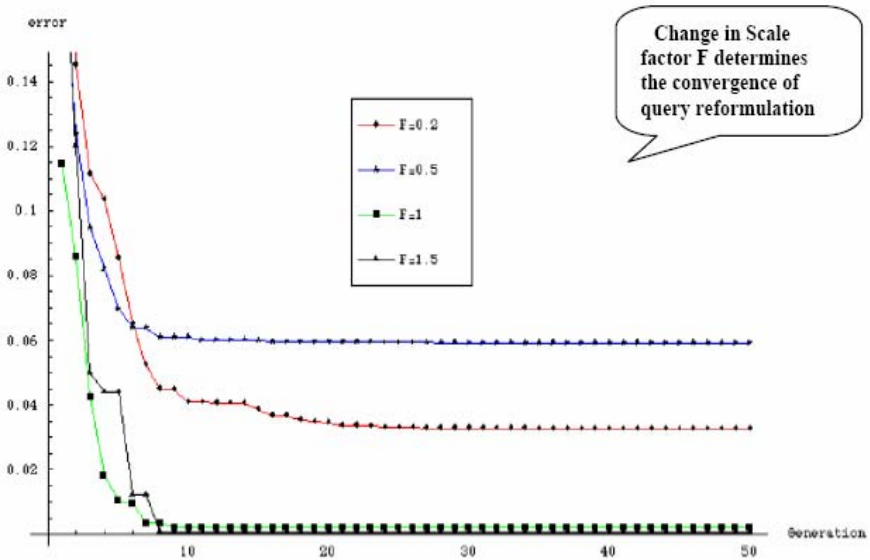


Fig. 3. Plot for Convergence of DE Under different Scale Factor in Query String

The result could be broadly compared with other popular clustering techniques, which clearly shows that *Differential Evolution* technique has been applied on any query point without having no prior knowledge on the query patten, Therefore functionally it works more faster than the other on line crawling mechanism.

5 Conclusion

This paper presents a query reformulation and clustering technique using *Differential Evolution*. The proposed *DE* automatically determines the type of a query and new pattern of query reformulation. Thus, effective search and crawling technique could be devised about the predictability of different complex query reformulation of user.

References

1. Baeza-Yates, R.: Graphs from search engine queries. In: van Leeuwen, J., Italiano, G.F., van der Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) SOFSEM 2007. LNCS, vol. 4362, pp. 1–8. Springer, Heidelberg (2007)
2. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Vigna, S.: Query suggestions using query-flow graphs. In: Proc. Of workshop on Web Search Click Data, WSCD 2009 (2009)
3. Radlinski, F., Joachims, T.: Query chains: learning to rank from implicit feedback. In: KDD 2005: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 239–248. ACM Press, New York (2005)
4. Lau, T., Horvitz, E.: Patterns of search: analyzing and modeling web query refinement. In: Proc. of Conf. on User modeling, UM 1999 (1999)
5. Rieh, S.Y., Xie, H.: Analysis of multiple query reformulations on the web: the interactive information retrieval context. *Inf. Process. Management.* 42(3), 751–768 (2006)
6. Gance, N.S.: Community search assistant. In: *Artificial Intelligence for Web Search*, pp. 91–96 (2001)
7. Craswell, N., Szummer, M.: Random walks on the click graph. In: Proc. of ACM SIGIR SIGIR 2007 (2007)
8. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., Vigna, S.: The query-flow graph: Model and applications. In: Proc. of ACM conf. on Inf. and Knowledge Manage. CIKM 2008 (2008)
9. Konar, A.: *Computational Intelligence: Principles, Techniques and Applications*. Springer, Berlin (2005)
10. Chou, H., Su, M.C., Lai, E.: A new cluster validity measure and its application to image compression. *Pattern Anal. Appl.* 7(2), 205–220 (2004)

On How Ants Put Advertisements on the Web

Tony White, Amirali Salehi-Abari, and Braden Box

School of Computer Science, Carleton University
1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6 Canada
{arpwhite,asabari,bbox}@scs.carleton.ca

Abstract. Advertising is an important aspect of the Web as many services rely on it for continued viability. This paper provides insight into the effectiveness of using ant-inspired algorithms to solve the problem of Internet advertising. The paper is motivated by the success of collaborative filtering systems and the success of ant-inspired systems in solving data mining and complex classification problems. Using the vector space formalism, a model is proposed that learns to associate ads with pages with no prior knowledge of users' interests. The model uses historical data from users' click-through patterns in order to improve associations. A test bed and experimental methodology is described, and the proposed model evaluated using simulation. The reported results clearly show that significant improvements in ad association performance are achievable.

Keywords: Ant Colony Optimization, Stigmergy, Pheromone, Collaborative Filtering.

1 Introduction

The proliferation of content on the World Wide Web (WWW) in the past decade has been, in the most part, economically supported by web advertising. A web advertisement is a section of a web page that is dedicated not to the content of the page but to graphics and/or text promoting the product or service of a third party. A web advertisement usually offers a link to an outside web site that provides further details regarding the product. The advertising party and the webmaster who owns the content have a contract which states that the advertiser will pay the webmaster in exchange for this service. The precise amount offered is frequently conditional on the success of the ad in compelling the end user to follow its link for more information. This success is usually measured by click-through rate (or CTR). A CTR is calculated by dividing the number of users who clicked on a specific ad by the number of times the ad is delivered.

It is then in the webmaster's best financial interest to maximize the number of his users that will, in fact, click-through. One approach to doing this is to ensure that the ad's content is matched to the users' interests, if there is a choice amongst candidate ads for a web page. It is this problem of choice that motivates the research reported in this paper. The anonymity of the web creates several obstacles to effectively resolving this problem of choice. It is not possible to track

who exactly is coming to your website without browser cookies or some other invasive technology, and even these are unreliable [1]. Even if it could be known which user is which, it is not easy to gauge their interests and demographics outside broad categories like geographic location or browser usage. A webmaster can only rely on one piece of information from the user when choosing which ad to match with a served page, and that is the content of the requested page.

Despite the limited amount of information, the webmaster must determine the users' interests, or at least which ads they are most likely to click on. Our main contribution includes the introduction of an ant-based algorithm for the association of web pages and ads with each other. Utilizing stigmergic [2,3] principles, the proposed algorithm builds models that can be quickly updated and provide recommendations for ad-serving. Moreover, we introduce a simulation test bed for evaluation of the proposed algorithm. The experiments performed demonstrate the utility of the proposed algorithm.

The paper consists of 6 further sections. The paper continues by providing important background information on Ant Colony algorithms in Section 2. Section 3 briefly describes related work in the area of ad association using biologically-inspired algorithms. Section 4 describes the main contributions of this paper: algorithms for ad association and a test bed that is used to evaluate them. Sections 5 and 6 describe the experimental setup and results respectively. Section 7 summarizes the key messages of the paper and briefly highlights potential future work.

2 Background

The heuristics that an ant colony uses to find food have inspired a computational metaheuristic that is known as Ant Colony Optimization (ACO) [4]. Starting with a simple, connected graph with start and destination nodes and every edge having a pheromone level, τ_{ij} , each ant steps from the node it is on to another connected node. The probability of the selection of an edge, e_{ij} , to follow at time t while the ant is located at node i can be calculated using Equation 1. Here, τ_{ij} is the amount of pheromone on edge e_{ij} and η_{ij} represents the desirability of a given direction. $N(i)$ contains the neighbors of node i . The parameters α and β are system parameters.

$$P_{ij}(t) = \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}}{\sum_{x \in N(i)} \tau_{ix}^{\alpha}(t)\eta_{ix}^{\beta}} \quad (1)$$

This process is repeated with each ant at each node until it reaches the destination. When the destination is reached an amount of pheromone is deposited on each edge that is inversely proportional to the total length of the path. To prevent premature convergence, pheromones are allowed to evaporate over time. Algorithmically, this means at each iteration, reduce the pheromone level, τ_{ij} , by multiplying it by $(1 - \rho)$ where $0 < \rho < 1$ is the evaporation rate. With this addition, we get the simplest form of ACO [4]. An ACO variant will be used to recommend Web advertisements and will be detailed in Section 4.1.

3 Related Work

The algorithms used for ad selection for Web pages are proprietary and remain largely unreported in the literature. Furthermore, *real time ad selection* is not based upon online click-through, which is the motivation for the research reported in this paper. However, the value of click-through data in this domain is well understood. See, for example, [5,6].

The problem of offline ad association can be viewed as a data mining problem if a large body of page requests can be used to induce classifiers. While the body of literature is large in this general space, Kim et al. [7] have used decision trees to guide the creation of advertisements for online storefronts and [6] has optimized search results using support vector machines and click-through data. We strongly believe that [5] could be used to analyze ads provided to our system to create the *adKeywords* vectors shown in the *clickThrough* algorithm (see Section 4) and facilitate the use of non-zero values of β (see Algorithm 3 and Table II). However, our interest in this paper is the incremental creation of classifiers online and, more specifically, through a use of biologically-inspired algorithms. With this latter qualification, prior research is sparse, with *AdPalette* [8] being noteworthy. *AdPalette* uses genetic algorithms to customize advertisements with usage, relying on crossover and mutation in order to combine promising ad components on pages.

4 Model

The research reported here was performed in a simulation environment. Figure 1 represents the actual Web environment being simulated. Simulated users with defined preferences create queries that are used to generate responses that contain a simulated advertisement. In the Web environment shown in Figure 1, users (e.g., Bob and Alice) interact with one or more web servers (e.g., Web Server A and B). When Bob asks for a page from Web Server A (indicated by 1 in the circle) content is returned that contains JavaScript that runs inside of Bob's browser. The JavaScript causes the Ad Server to be contacted with keywords extracted from the content delivered by Web Server A. The ad returned from the Ad Server is shown by the number 3 in a circle in Figure 1.

The simulation models the interaction that occurs between the web page users and the web server that processes page requests and matches the advertisements to the content. Two types of processes run simultaneously: one server script where most of the actual computing occurs, and a *user script* that contains randomly generated users that create and send off page requests to the server to be processed. Essentially, the user script models a user's interactions with various web servers that are connected, in turn, to an ad serving system. The functions of the user script are to (a) generate users and (b) generate queries and assess responses. The *server script* represents the functions of the ad serving system. The server script is responsible for analyzing the queries that it receives and making a decision as to which ad to serve.

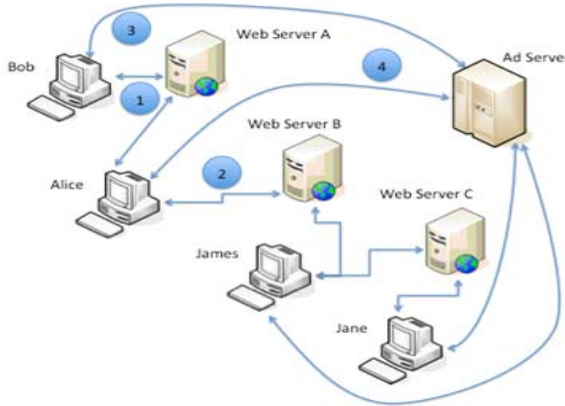


Fig. 1. The Ad Placement Scenario

The first function of the user script is to generate up to l users. Each user entity is meant to simulate one actual user’s preferences and choices. To represent the variability of users’ interests, each user has a rating for each of a pool of m keywords that represents how interested the user is in that particular subject. Note, since we are merely simulating a user, actual keywords are not used but the preferences, $userInterest$, recorded as a vector of values $[k_0, k_1, \dots, k_{m-1}]$, with the value of each k_i being between 0 – meaning no interest in the subject i – and 1 – meaning the highest possible interest in i – are instead. These values are randomly generated, but are weighted towards the extremes such that there is a one third chance of the interest level being between 0.75 and 1, and a one third chance of the interest level being between 0 and 0.25. This is meant to represent that is more likely that a person has a considerable amount of interest in a topic or very little interest at all, as opposed to being ambivalent about it.

After the user is created, the user script synthesizes finding a page of interest, defined by keywords, to the user. Pages are represented by randomly generating a relevance vector $[r_0, r_1, \dots, r_{m-1}]$ to represent a possible page, again with values between 0 and 1 that represent how tightly tied to the subject matter the content is. In an actual implementation with real Web content, this could be determined using a vector space model [9] that analyzes the page content relevant to other pages. Since users do not randomly surf pages, but go to pages that match their interests, only pages with keywords relevant to the user are of interest. To simulate this, the angle between the interest vector of the user and the relevance vector of the page is first determined using Equation 2.

$$angle(a, b) = \arccos \left(\frac{\sum_{i=0}^m a_i b_i}{\|a\| \|b\|} \right) \tag{2}$$

In Equation 2, a and b are vectors of size m with $\|a\|$ and $\|b\|$ being the lengths of the vectors a and b respectively. If the angle computed for the page is greater than some chosen threshold, R_{thresh} , it is rejected. If the page is rejected, then each of

Algorithm 1

```

generatePage(Array userInterest)
  page = new Array[m]
  for i = 0 to m - 1 do
    page[i] = random value from -0.25 to 1.25
    if page[i] > 1 then
      page[i] = page[i] - 0.25
    end if
    if page[i] < 0 then
      page[i] = page[i] + 0.25
    end if
  end for
  angle = angle between page and userInterest using Equation 2
  while angle >  $R_{thresh}$  do
    for i = 0 to m - 1 do
      20% chance of:  $page[i] = \frac{page[i] + userInterest[i]}{2}$ 
    end for
    angle = angle between page and userInterest using Equation 2
  end while
  return page

```

the m values has a 20% chance of being averaged with the user's interest values. This continues in a loop until the threshold is met. When the *generatePage* algorithm (see Algorithm 1) completes it is guaranteed that the generated page represents the interests of the modeled user; however, there may be content unrelated to the user's primary interests.

When the server receives a page from a user, its function is to decide which ad should be matched with the given page. This is achieved by using the Ad Association (A2) algorithm as defined in Section 4.1. Finally, the page will be sent back to the appropriate user with one of n possible ads attached. The user then evaluates the ad, and determines whether to click it. In a real-life scenario, the person receiving the ad would judge the ad on his own using preferences and goals, and choose whether the ad has piqued his interest enough to click it. However, it would be incredibly hard to simulate a human in this way, so it has been simplified as follows. Each ad has a number of keywords associated with it, denoted by pm , of the m possible keywords ($pm < m$). The more interest a user has in the pm keywords, the more likely the ad is to be a success. The chance of success is determined on a logarithmic scale. For example, with 10 keywords associated with an ad, there is an expected value of 1/100 th chance of a click-through, but varying from 1/10,000 th chance if the user has no interest at all in the ad's keywords and guaranteed success if the user has maximum interest in all keywords. The *clickThrough* algorithm below indicates how success is computed, with the *adKeywords* array containing 0 for completely irrelevant keywords not thought to be useful for the ad and 1 for keywords that are considered important or completely relevant. The *adKeywords* array therefore contains pm entries that are 1 and $m - pm$ entries that are 0. The *normalization* value is a constant for the system.

Algorithm 2

```

clickThrough(Array adKeywords, Array userInterest)
total = 0
for i = 0 to n - 1 do
  if adKeywords[i] == 1 then
    total = total + userInterest[i]
  end if
end for
return  $10^{(total-pm)*normalization}$ 

```

Finally, when the success of the ad is determined, information is once again passed back to the server. The server executes the A2 algorithm and makes changes based on the page relevance vector, the ad selected and the user's choice in regard to the ad. This whole process repeats many times; multiple users each making a series of page requests.

4.1 Ad Association (A2) Algorithm

In this ant-inspired algorithm, each ad is a given fixed path with m nodes, one for each of the possible interest keywords along with the *adKeywords* described in the previous section. When an ad is served to the user, and is successful, then it positively reinforces that ad's path. If it fails, it negatively reinforces the path.

There are 3 parts to the A2 Algorithm: the model, a method for choosing the best ad considering a page input, and a method for changing the model. The model, M , is a collection of one vector per ad, a , and each vector has a value for each of the m interest keywords, i.e., $M = \{v_{01}, v_{02}, \dots, v_{0m-1}, v_{11}, \dots, v_{n-1m-1}\}$. Each of these nodes, v_{ij} , initially contains a value, τ_0 , just slightly above zero representing the amount of pheromone on that part of the path. There are n ads that can be served. Furthermore, each ad has pm defining keywords, these being used to decide on whether a click-through occurs or not. The pm keywords are chosen from the m possible keywords.

As the server receives page requests from users, the A2 algorithm (See Algorithm 3) executes for the purpose of determining which of the n ads should be returned. The algorithm proceeds by comparing the page's relevance vector with each of the n ads by measuring the angle between the two vectors using Equation 2. This comparison has two components: first, the feedback gathered from previous ad associations and second, the comparison between the page and the ad keywords that are considered relevant. The smaller the angle, the more similar the ad's vector is to the relevance vector. The server chooses which ad to return randomly biased by rank. There is a $\frac{1}{2^r}$ chance that the r^{th} best ad is returned to the user, thus ensuring a heavy bias towards the best ads; i.e., the best ad will only be returned half of the time. Once the server sends out the ad, it waits for information on whether the user clicked the ad. If the ad was successful, the *recordAd* algorithm executes. This algorithm ensures that the vector corresponding to that successful ad, i , is updated so that it increases its pheromone values for all keywords in i , but increases the page's more relevant

Algorithm 3. Ad Association (A2) Algorithm

```

decideOnAd(Array page)
  angles = new Array[n]
  for a = 0 to n - 1 do
    angleToAd = angle between page and ads[a] using Equation 2
    angleToV = angle between page and v[a] using Equation 2
    angles[a] = (angleToAdβ) × (angleToVα)
  end for
  angles = sortInIncreasingOrder(angles)
  index = 0
  while true do
    if index >= n then
      return last ad in angles array
    else
      50% chance of returning ad with angle at index
      Otherwise: index = index + 1
    end if
  end while

recordAd(adNumber, Array page, success)
if success == true then
  for i = 0 to m - 1 do
    v[adNumber][i] = v[adNumber][i] + page[i] * c
  end for
else
  for i = 0 to m - 1 do
    v[adNumber][i] = v[adNumber][i] - page[i] * k
  end for
end if

```

keywords more, as shown in Equation 3. The relevance vector is r , and c is a constant that controls how much each success influences the model, $0 < c < 1$. This is equivalent to an ant spreading pheromone on the path to the successful ad, and making that path more appealing to future ants with similar vectors that match.

$$\forall j \quad v_{ij}^{t+1} = v_{ij}^t + r[j] \times c \quad (3)$$

To counter the pheromone values from growing out of control, values are bounded and the model also adjusts them when an ad fails. In a real-life application, it is much harder to tell when an ad fails as no message can be sent to the server saying that the user did not do some action. To circumvent this, it is assumed that there is a timeout function such that if the ad is not successful in a given time frame, then it counts as a failure. In the simulation, the user script just returns a “failure” result. When the server receives a failure, it reduces all pheromone values for the associated ad, i , but reduces it more for the keywords most relevant to the page in question. This effect is shown in Equation 4.

where k is a constant that controls how much each failure influences the model, $0 < k < 1$ and $k \ll c$. Here, k has the role of evaporation in ACO. τ_0 is a very small number, $\tau_0 \ll k$.

$$\forall j \quad v_{ij}^{t+1} = v_{ij}^t + \max(r[j] \times k, \tau_0) \quad (4)$$

The purpose of having this pheromone reduction is to prevent one ad from being overused early on, and becoming dominant before other ads get a chance to demonstrate relevance. Equations 3 and 4 form the basis of the *recordAd* algorithm.

5 Experiments

Multiple trials were completed for the A2 algorithm to ensure consistency and result reproducibility. Three trials are reported in Figure 2 in order to show the typical variability in simulated performance.

Table 1. Experimental Parameters

| Variable | Value | Variable | Value |
|----------------------|------------|----------|-------|
| R_{thresh} | 20° | k | 0.01 |
| τ_0 | 10^{-6} | c | 0.999 |
| n | 20 | pm | 10 |
| m | 100 | l | 100 |
| <i>normalization</i> | 0.4 | α | 1 |
| Total page requests | 500,000 | β | 0 |

The values of c and k were chosen to reflect the relative probabilities of success and failure respectively; the ratio being approximately 100 to 1 for the scenario modeled here. The value of β were chosen to reflect an extremely pessimistic scenario in which nothing was known about the keywords associated with the advertisements to be presented. In essence, this value of β says that we know nothing about the contents of the ad, which may be true if the ad is an image or video or the provider seeks to provide a “black box” ad. That said, this is an extreme scenario and was chosen in order to test whether the system could improve based purely upon observed feedback from users.

The value of the *normalization* coefficient was chosen to ensure that the average chance of click-through for a page with half of the expected keywords correct would be 0.01, with the range of probabilities varying from 1 (for all expected keywords correct) to 0.0001 (for no keywords correct).

The page requests were broken down into 50 blocks of 10,000 requests each. For each block, the average success rate as determined by the user was measured. Also measured was the average expected value of the success rate if the ads were chosen randomly. This is shown as a horizontal line in Figure 2.

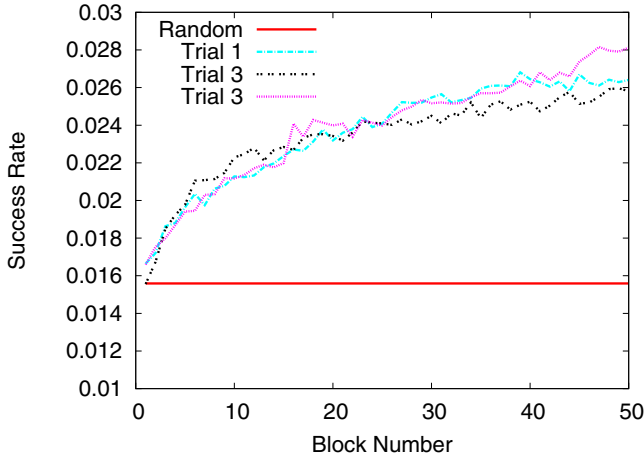


Fig. 2. Success Rate Variation with Block Number

6 Results and Discussion

The data gathered from testing the A2 Algorithm is promising. As shown in Figure 2 the average chance of ad success will be increased over time. Over the course of 500,000 page requests, there was a 70% to 80% improvement in efficiency over the random pairing of ads when viewed across the 3 trials reported here.

Note that, the proposed algorithm does not always return the best match from its model, but returns one from the top few ads with very high frequency (87.5% from the top 3 ads). This is important and deliberate. It was found during the implementation and associated experimentation that without this feature, the model would converge too quickly on one ad that happened to show promise early on. Success would breed more recommendations for that ad, which would only increase the likelihood of it being recommended again later. Adding the chance that any ad could be picked kept the model from being dominated too early by any particular ad. Thus, robustness was maintained.

As can be observed in Figure 2, the system continues to learn even after 500,000 page requests making it likely that 100% improvement over initial system performance is achievable. Combining the clear benefits that this algorithm produces in terms of advertising success and the ease with which it handles large quantities of data makes it attractive.

7 Conclusions

This paper provides insight into the effectiveness of using an ant-based algorithm to improve Internet advertising. A webmaster could use the proposed A2 algorithm with minor modifications. As shown with an artificial environment, it should be able to increase ad success, and therefore, advertising revenue by

over 70% after processing a reasonable amount of traffic for a large website. If it were used, with refinements made to adjust for real-life variables and efficiencies added to reflect issues unique to the website, it would be a simple piece of software with potential revenue benefits. It is worth looking further into using the A2 algorithm as a predictor of advertisement effectiveness. We believe that click-through data mining techniques as described in [5] and [6] (as examples) could provide a valuable starting point for system ad keyword initialization thereby allowing for better-than-random initial system performance. Furthermore, larger data sets should be tested beyond the small problem space that is used here.

Beyond this, the next stage would be to implement the algorithm on a real web server. It would run continuously and intercept real incoming customer data and produce actual ads, using vector-space models to convert the requested pages into the vectors used by the algorithm. It could be further refined to take into account that some advertisements may pay more to be shown, that some ads have different sizes and page positioning from others and that more than one ad is often shown on a site at once.

References

1. Kristol, D.M.: Http cookies: Standards, privacy, and politics. *ACM Trans. Internet Technol.* 1(2), 151–198 (2001)
2. Dorigo, M., Bonabeau, E., Theraulaz, G.: Ant algorithms and stigmergy. *Future Gener. Comput. Syst.* 16(9), 851–871 (2000)
3. Ramos, V., Abraham, A.: Swarms on continuous data. In: *The 2003 Congress on Evolutionary Computation*, pp. 1370–1375 (2003)
4. Dorigo, M., Stützle, T.: *Ant Colony Optimization*. Bradford Book (2004)
5. Joachims, T., Radlinski, F.: Search engines that learn from implicit feedback. *Computer* 40, 34–40 (2007)
6. Joachims, T.: Evaluating retrieval performance using clickthrough data. In: *SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval* (2002)
7. Kim, J.W., Lee, B.H., Shaw, M.J., Chang, H.L., Nelson, M.: Application of decision-tree induction techniques to personalized advertisements on internet storefronts. *International Journal of Electronic Commerce* 5(3), 45–62 (2001)
8. Karuga, G.G., Khraban, A.M., Nair, S.K., Rice, D.O.: Adpalette: an algorithm for customizing online advertisements on the fly. *Decision Support Systems* 32(2), 85–106 (2001)
9. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Commun. ACM* 18(11), 613–620 (1975)

Mining Association Rules from Semantic Web Data

Victoria Nebot and Rafael Berlanga

Universitat Jaume I, Campus de Riu Sec, E-12071 Castellón de la Plana, Spain
{romerom,berlanga}@lsi.uji.es

Abstract. The amount of ontologies and semantic annotations available on the Web is constantly increasing. This new type of complex and heterogeneous graph-structured data raises new challenges for the data mining community. In this paper, we present a novel method for mining association rules from semantic instance data repositories expressed in RDF/S and OWL. We take advantage of the schema-level (i.e. *Tbox*) knowledge encoded in the ontology to derive just the appropriate transactions which will later feed traditional association rules algorithms. This process is guided by the analyst requirements, expressed in the form of a query pattern. Initial experiments performed on real world semantic data enjoy promising results and show the usefulness of the approach.

Keywords: Semantic Web, Data Mining, Instance Data, Association Rules.

1 Introduction

Thanks to the standardization of the ontology languages RDF/S¹ and OWL², the Semantic Web has been realized and the amount of available semantic annotations is ever increasing. This is due in part to the active research concerned about learning knowledge structures from textual data, usually referred as Ontology Learning [1]. However, little work has been directed towards mining from the Semantic Web. We strongly believe that mining Semantic Web data will bring much benefit to many domain-specific research communities where relevant data are often complex and heterogeneous, and a large body of knowledge is available in the form of ontologies and semantic annotations. This is the case of the clinical and biomedical scenarios, where applications often have to deal with large volumes of complex data sets with different structure and semantics. In this paper, we investigate how ontological instances expressed in OWL can be combined into transactions in order to be processed by traditional association rules algorithms, and how we can exploit the rich knowledge encoded in the respective ontologies to reduce the search space.

The rest of the paper is organized as follows. Section 2 gives an overview of the related work. Section 3 explains the basics of the two integrated technologies,

¹ RDF/S: <http://www.w3.org/TR/rdf-concepts/rdf-schema/>, '04

² OWL: <http://www.w3.org/TR/owl-features/>, '04.

association rules mining and OWL DL ontologies and motivates the problem with a running example. Section 4 contains the general methodology and foundations of the approach. Section 5 shows the experimental evaluation and Section 6 gives some conclusions and future work.

2 Related Work

Most research on data mining for semantic data is based on Inductive Logic Programming (ILP) [2], which exploits the underlying logic encoded in the data to learn new concepts. Some examples are presented in [3] and [4]. However, there is the inconvenient of rewriting the data sets into logic programming formalisms and most of these approaches are not able to identify some hidden concepts that statistical algorithms would.

Other studies extend statistical machine learning algorithms to be able to directly deal with ontologies and their associated instance data. In [5], a framework is presented for designing kernel functions that exploit the knowledge of the underlying ontologies. Recently, in [6] and [7] new frequent association rules algorithms are proposed which make use of similarity functions in a similar way to the previous kernel functions.

A recent trend in data mining research is to consider more complex and heterogeneous structures than single tabular data, mainly tree and graph structured data. In this line, we can find frequent subtree [8] and graph mining [9], whose aim is to identify frequent substructures in complex data sets. Albeit interesting, these algorithms do not serve the purpose of finding interesting content associations in RDF/S and OWL graphs because they are concerned with frequent syntactic substructures but not frequent semantically related contents. Indeed, frequent graph substructures usually hide interesting associations that involve contents represented with different detail levels of the ontology. Moreover, although the underlying structure of RDFS and OWL is a graph, reasoning capabilities must be applied to handle implicit knowledge.

Finally, we find some work aimed at integrating knowledge discovery capabilities into SPARQL [3] by extending its grammar. Some examples are [10], which can be plugged with several data mining algorithms and [11], which finds complex path relations between resources. Inspired by these works, we have also extended SPARQL grammar to define association rule patterns over the ontological data but in a less restrictive way than the one imposed by SPARQL. These patterns allow the system to focus only on the interesting features, reducing both the number and length of generated transactions.

3 Preliminaries

The problem of discovering association rules was first introduced in [12]. It can be described formally as follows. Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of m literals,

³ SPARQL: <http://www.w3.org/TR/rdf-sparql-query>, '08

called items. Let $D = \{t_1, t_2, \dots, t_n\}$ be a database of n transactions where each transaction is a subset of I . An itemset is a subset of items. The support of an itemset S , denoted by $sup(X)$, is the percentage of transactions in the database D that contain S . An itemset is called frequent if its support is greater than or equal to a user specified threshold value.

An association rule r is a rule of the form $X \Rightarrow Y$ where both X and Y are nonempty subsets of I and $X \cap Y = \emptyset$. X is the antecedent of r and Y is called its consequent. The support and confidence of the association rule $r : X \Rightarrow Y$ are denoted by $sup(r)$ and $conf(r)$.

The task of the association data mining problem is to find all association rules with support and confidence greater than user specified minimum support and minimum confidence threshold values [12].

DLs allow ontology developers to define the domain of interest in terms of *individuals*, *atomic concepts* (called *classes* in OWL) and *roles* (called *properties* in OWL). Concept constructors allow the definition of *complex concepts* composed of atomic concepts and roles. OWL DL provides for union (\sqcup), intersection (\sqcap) and complement (\neg), as well as enumerated classes (called *oneOf* in OWL) and existential (\exists), universal (\forall) and cardinality ($\geq, \leq, =$) restrictions involving an atomic role R or its inverse R^- . In OWL DL it is possible to assert that a concept C is subsumed by D ($C \sqsubseteq D$), or is equivalent to D ($C \equiv D$). Equivalence and subsumption can be also asserted between roles and roles can have special constraints (e.g., transitivity, symmetry, functionality, etc.) Regarding instance axioms, we can specify the class C of an instance a ($C(a)$), or the relations between two instances a and b ($R(a, b)$). A DL ontology consists of a set of axioms describing the knowledge of an application domain. This knowledge ranges over the terminological cognition of the domain (the concepts of interest, its *Tbox*) and its assertions (the instances of the concepts, its *Abox*).

Fig. 1 shows a fragment of the *Tbox* of a DL ontology designed for patients with arthritis-related diseases. Through semantic annotation, clinicians can annotate data sets with ontology terms and relationships from the axioms in Fig. 1, creating a repository of semantic annotations in OWL (an *Abox*) which must be consistent with the ontology axioms (*Tbox*). The right hand side of Fig. 1 shows an excerpt

| Axioms | subject | predicate | object |
|---------------------------------------------------|------------------|--------------|------------------|
| $Patient \sqsubseteq \exists doB.string$ | PTNXZ1 | doB | "20021108" |
| $Patient \sqsubseteq \exists sex.Gender$ | PTNXZ1 | sex | Female |
| $Patient \sqsubseteq \exists hasReport.Report$ | PTNXZ1 | hasReport | RPT1 |
| $Report \sqsubseteq \exists hasDiag.Disease$ | RPT1 | hasDiag | PolyArthritis |
| $Report \sqsubseteq \exists hasSection.Section$ | RPT1 | hasSection | STreat1 |
| $hasReport \sqsubseteq belongsTo^-$ | STreat1 | type | Treat |
| $Treat \sqsubseteq Section \sqcap$ | STreat1 | hasDrug | Methotrexate |
| $\quad \quad \quad \exists hasDrug.Drug$ | PTNXZ1 | ... | MH1 |
| $Patient \sqsubseteq \dots MotherHist$ | MH1 | hasDiagnosis | RheumatoidArthr. |
| $MotherHist \sqsubseteq \exists hasDiag.Disease$ | MH1 | treatedWith | NSAIDS |
| $MotherHist \sqsubseteq \exists treatedWith.Drug$ | PolyArthritis | type | Arthritis |
| $RheumaticDis. \sqsubseteq Disease$ | RheumatoidArthr. | type | Arthritis |

Fig. 1. Ontology axioms (Tbox) and semantic annotations (Abox)

of the semantic annotations associated to a patient named *PTNXZ1*. Notice that semantic annotations are represented as triples (*subject, predicate, object*). In this paper, we separate the *Tbox* from the *Abox* for practical issues. In general, we use the term *ontology* to refer to the *Tbox* and *instance store* to refer to the *Abox*. Thus, our data mining problem is defined as follows:

Data Mining Problem: Given an OWL instance store *IS* consistent with the ontology *O* and a mining pattern *Q* expressed in the extended SPARQL syntax (see Section 4.1), find association rules from *IS* according to the mining pattern *Q* with minimum support and confidence threshold values.

The previous data mining problem can be thought as a classic data mining problem where the input data must be derived from the ontology in order to generate transactions according to the user specification.

4 Methodology

In this section we present a detailed view of our method along with the definitions that sustain it. Fig. 2 depicts a schematic overview of the whole process. The user specifies a mining pattern following an extended SPARQL syntax. Then, the transaction extractor is able to identify and construct transactions according to the mining pattern previously specified. Finally, the set of transactions obtained are processed by a traditional pattern mining algorithm, which finds association rules of the form specified in the mining pattern with support and confidence greater than user's specified ones.

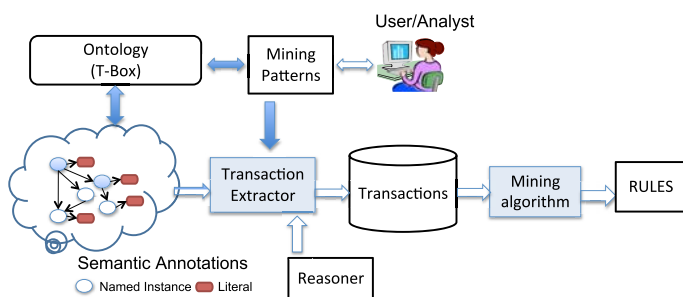


Fig. 2. Architecture of our proposal for mining semantic annotations

4.1 Mining Pattern Specification

The user has to specify the kind of patterns (s)he is interested in obtaining from the repository. Since semantic annotations are encoded in RDF/S and OWL, we have extended SPARQL with a new statement that allows to specify a mining pattern. The syntax is inspired by the Microsoft Data Mining Extension (DMX),

```

[1] Query ::= Prologue( SelectQuery | ConstructQuery | DescribeQuery | AskQuery |
MiningQuery )
[2] MiningQuery ::= CREATE MINING MODEL' Source '{' Var 'RESOURCE' 'TARGET' (
Var ( 'RESOURCE' | 'DISCRETE' | 'CONTINUOUS' )
'MAXCARD1'? 'PREDICT'? 'CONTEXT'?)+ '}'
DatasetClause* WhereClause SolutionModifier UsingClause
[1.2] UsingClause ::= 'USING' SourceSelector BrackettedExpression

```

Fig. 3. Extended SPARQL grammar for the CREATE MINING MODEL statement

```

CREATE MINING MODEL <http://krono.act.uji.es/patients_repository>
{ ?patient RESOURCE TARGET
  ?drug RESOURCE
  ?disease RESOURCE PREDICT
  ?report RESOURCE CONTEXT
}
WHERE
{ ?patient rdf:type Patient .
  ?drug rdf:type Drug .
  ?disease rdf:type Disease .
  ?report rdf:type Report .
}
USING apriori (SUPPORT = 0.05, CONFIDENCE = 0.07)

```

Fig. 4. Example of extended SPARQL query with CREATE MINING MODEL statement

which is an SQL extension to work with data mining models in Microsoft SQL Server Analysis Services.⁴

The extended SPARQL grammar is depicted in Fig. 3, and Fig. 4 shows an example query. We extend the SPARQL grammar rule *Query* by adding a new symbol, named *MiningQuery*. This symbol expands to the keywords CREATE MINING MODEL followed by the *Source*, which identifies the input repository. The body consists of variables the user is interested in mining. Next to each variable, we specify its content type: RESOURCE for variables holding RDF resources, DISCRETE for variables holding literal values and CONTINUOUS for variables holding a continuous literal value. In case we want to find patterns with just one occurrence of the variable, we attach the keyword MAXCARD1 to the variable. By default, patterns found can contain more than one occurrence of each variable. Moreover, we specify the consequent of the rule by attaching the keyword PREDICT. Finally, the keyword TARGET denotes the resource under analysis, which must be an ontology concept. The analysis target determines the set of obtained rules. In the query example, the analysis target is a *Patient*. In the WHERE clause, we specify the restrictions over the previous variables. The good news is that we do not expect users to have an exact knowledge of the ontology structure. Therefore, users do not have to input the paths relating the pattern variables in SPARQL. Instead, they are asked to specify just the type (i.e. ontology concept) that the variables refer to. In case variables are not resources, users must specify the type of the domain of that value followed by the data type property. From now on, we refer to these ontology concepts selected

⁴ DMX Reference: <http://technet.microsoft.com/en-us/library/ms132058.aspx>

by the user as the *features* set. For example, in Fig. 4 the user is interested in obtaining which drugs are associated to which diseases, so the features set is $\{Drug, Disease\}$. Finally, the *UsingClause* grammar symbol defines the name and parameters of the learning algorithm.

Since we do not ask the user to specify the exact relations, the previous query model introduces some ambiguity regarding the items that form a transaction. When the user specifies the mining pattern, (s)he is thinking about obtaining subsets of drugs that are frequently administered to patients having a certain disease. Therefore, (s)he restricted the features set to be of type *Drug* and *Disease*, respectively. However, these concepts may appear not only under the user intended context but all over the ontology. That is to say, the same conceptual entities may appear under different contexts, making it challenging for the system to automatically discover what the users' intentions really are. As previously mentioned, the user could remove this ambiguity by specifying in the SPARQL extended query the exact relation of concepts in the ontology through pattern graph triples in the WHERE clause. However, this task can be cumbersome and not always viable. In this paper, we want to relief the user from this burden and let the system handle the task of finding appropriate contexts. Thus, in order to provide the right sense to the query, user can select the intended context with the CONTEXT keyword attached to the appropriate concept.

4.2 Transaction Extractor Foundations

Since our goal is to be able to identify and construct transactions according to the user's mining pattern, in this section we present all the definitions that sustain the method we have developed.

Definition 1. Let O be an ontology, IS an instance store consistent with O and C_T the analysis target. The target instances are the set $I_T = \{i/i \in IS, O \cup IS \models C_T(i)\}$.

In the running example C_T is *Patient* and I_T is the set of all instances classified as *Patient*.

Definition 2. $Path(C, C') = (r_1 \circ \dots \circ r_n) \in Paths(C, C')$ is an aggregation path from concept C to concept C' of an ontology O iff $O \models C \sqsubseteq \exists r_1 \circ \dots \circ r_n.C'$.

Definition 3. Let O be an ontology, C_T the analysis target and C_a, C_b two named concepts.

$Contexts(C_a, C_b, C_T) = \{C''/C'' \sqsubseteq C'\}$ are least common reachable concepts and their subconcepts. That is,

- (1) $\exists p_1 \in Paths(C_a, C') \wedge \exists p_2 \in Paths(C_b, C') \wedge \exists p_3 \in Paths(C', C_T)$ (C' is common reachable concept).
- (2) if $\exists p_x \in Paths(C_a, E) \wedge \exists p_y \in Paths(C_b, E)$ then $\exists p_z \in Paths(C', E)$ (C' is least).

Example 1. The ontology fragment on the left in Fig. 5 models some part of the patient’s medical record. In this example we can infer that $Contexts(Disease, Drug, Patient) = \{Report, MotherHistory\}$.

Definition 4. Let i and i' be two named instances of an instance store IS . $Path(i, i') = (r_1 \circ \dots \circ r_n) \in Paths(i, i')$ is an aggregation path from instance i to instance i' in the instance store IS .

Definition 5. Let $i_T \in I_T$ be a target instance and i_a, i_b two named instances in an instance store IS .

$Contexts(i_a, i_b, i_T) = \{i'\}$ are least common reachable instances. That is,

- (1) $\exists p_1 \in Paths(i_a, i') \wedge \exists p_2 \in Paths(i_b, i') \wedge \exists p_3 \in Paths(i', i_{SUB})$ (i' is common reachable instance).
- (2) if $\exists p_x \in Paths(i_a, i'') \wedge \exists p_y \in Paths(i_b, i'')$ then $\exists p_z \in Paths(i', i'')$ (i' is least).

Example 2. The right hand side of Fig. 5 shows an example of instance store represented as a graph that is consistent with the ontology fragment. In this example, we can infer that $Contexts(PolyArthritis, Methotrexate, PTN_XY21) = \{RPT_NH23\}$.

Definition 6. Let O be an ontology and IS an instance store consistent with O . Two instances i_a, i_b , $C(i_a) \neq C(i_b)$, $C(i_a), C(i_b) \in features$ belong to the same transaction under a target instance i_T iff i_a, i_b are context compatible. That is, $\exists C_1 \in \{C(i_x)/i_x \in Contexts(i_a, i_b, i_T)\}$, $\exists C_2 \in Contexts(C(i_a), C(i_b), C_T)$ such that $O \models C_1 \sqsupseteq C_2$ where $C(i)$ is the asserted class for instance i in IS .

Example 3. In the right hand side of Fig. 5, instances *Methotrexate* and *PolyArthritis* are context compatible. That is, their context is *Report* both at the instance level (Abox) and at the conceptual level (Tbox).

As previously mentioned, context compatible instances may appear under very different contexts in an ontology, making it hard for the system to guess the user intended context. Currently, we allow users to specify the contexts of interest directly in the query. The concept denoting the selected context is denoted as C_{CTXT} . If no context is specified, we assume $C_{CTXT} = C_T$. Now, transactions can be unambiguously defined as follows.

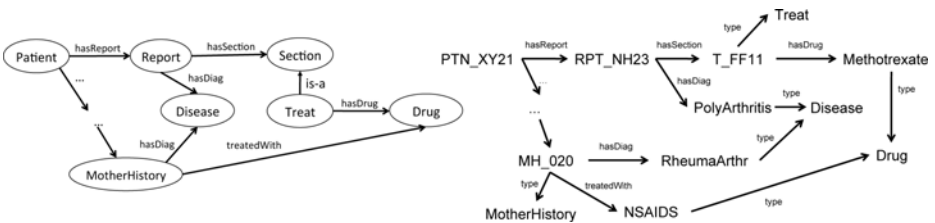


Fig. 5. Ontology graph and instance store fragment

Definition 7. Let O be an ontology and IS an instance store consistent with O . An instance transaction associated to target instance i_T under context C_{CTX_T} is a tuple of instances (i_1, i_2, \dots, i_n) such that $\forall i_x, 1 \leq x \leq n, O \cup IS \models C(i_x)$ where $C \in \text{features}$ and $\forall (j, k), 1 \leq j, k \leq n$, if $C(i_j) \neq C(i_k), (i_j, i_k)$ are context compatible under i_T w.r.t. the user selected context C_{CTX_T} .

Example 4. Given $C_T = \text{Patient}$ and $\text{features} = \{\text{Drug}, \text{Disease}\}$ we have three options. If the user does not select a context (i.e. $C_{CTX_T} = \text{Patient}$) the transaction $\{\text{Methotrexate}, \text{PolyArthritis}, \text{RheumatoidArthritis}, \text{NSAIDS}\}$ would be generated. If $C_{CTX_T} = \text{Report}$, the transaction $\{\text{Methotrexate}, \text{PolyArthritis}\}$ would be generated. Finally if $C_{CTX_T} = \text{MotherHistory}$, the generated transaction would be $\{\text{RheumatoidArthritis}, \text{NSAIDS}\}$.

5 Evaluation

The current implementation of the transaction extractor has been developed on the top the ontology indexing system proposed in [13], which also provides a simple reasoning mechanism over the ontology indexes. In order to show the usefulness of our proposal, we test the method over a real-world instance store holding OWL annotations about patient's follow-ups. These annotations have been generated in the context of the Health-e-Child project⁵, and they are consistent with an ontology similar to the one used as example in Fig. 1. The semantic annotations contain information about 588 patients with very heterogeneous structure. The total number of semantic annotations is 629.000, which gives more than 1000 semantic annotations per patient on average.

Table 1 shows some query pattern examples⁶ along with the user selected context, the number of transactions generated by our method and the number of rules generated with the *Apriori* algorithm [14] for mining association rules. The query patterns specify the ontology concepts acting as interesting features, leaving the PREDICT attribute unmarked. As it can be observed, the first query is executed in for the context of *Patient*, which means a transaction will be generated for each patient holding features of type *Disease*, *Drug* and *Finding*. The selection of the context is crucial because it determines both the number and contents of each transaction, and therefore, the obtained rules. Notice the number of transactions is very reduced thanks to the context selection and, consequently, the number of generated rules.

Table 1 also shows some examples of the rules generated from the previous queries. The obtained rules are very clear and useful thanks to the query pattern restrictions and latter transaction generation, which extremely reduces the features' search space thus, the complexity and overload produced by uninteresting features. To corroborate this fact, we generated transactions with all the possible features at the context of *Patient* and the *apriori* algorithm obtained more than 400.000 rules, of which the first hundred were uninteresting. Notice

⁵ <http://www.health-e-child.org/>

⁶ We omit full syntax of the query pattern due to space restrictions.

Table 1. Examples of rules obtained with different queries and contexts (between brackets). All the queries are performed with minimum confidence of 0.7. LOM stands for limitation of motion, ANA for anti-nuclear antibody and RF for Rheumatoid Factor.

| Query | Examples of rules | Sup. | Conf. |
|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|-------|-------|
| <i>Drug</i> \wedge <i>Finding</i> \implies <i>Disease</i> [Patient] 225 transactions 22 rules (sup=0.1) | LargeJointsAffected \implies Oligoarthritis | 0.19 | 0.91 |
| | PresenceOfANA \implies Oligoarthritis | 0.14 | 0.76 |
| | PrevDrugEtanercept \implies PrevDrugMethotrexate | 0.12 | 0.9 |
| | PrevDrugPrednisone \implies SystemicArthrities | 0.11 | 1.0 |
| | Fever , Rash \implies SystemicArthrities | 0.11 | 1.0 |
| <i>Finding</i> \implies <i>Disease</i> [Report] 492 transactions 12 rules (sup=0.05) | LargeJointsAffected \implies Oligoarthritis | 0.087 | 0.91 |
| | PresenseOfANA \implies Oligoarthritis | 0.063 | 0.76 |
| | Fever , Rash \implies SystemicArthrities | 0.05 | 1.0 |
| <i>Finding</i> \implies <i>Finding</i> [Med.Ev.] 84 transactions 21 rules (sup=0.08) | ANA_Negative \implies RF_Negative | 0.36 | 0.86 |
| | LOMRightShoulder \implies LOMRightWrist | 0.08 | 0.81 |
| | LOMLeftHip \implies LOMRightHip | 0.08 | 0.73 |

for queries 1 and 2 we get the same rule stating that the presence of ANA implies oligoarthritis disease. However, the support is different because their respective contexts are different, which generates different transaction sets.

Overall, by specifying a query pattern and selecting the intended context, the features' search space is extremely reduced. That is, the generated transactions contain only the interesting stated and inferred features for the user at the level of granularity specified in the context. Therefore, this process eases the task of the association rule algorithm, which extracts small and very useful subsets of interesting rules.

6 Conclusions

We have presented a novel method for mining association rules from heterogeneous semantic data repositories expressed in RDF/S and OWL. To the best of our knowledge, this problem has only been considered to a minor extend. The intuition under the method developed is to extract and combine just the interesting instances (i.e. features) from the whole repository and flatten them into traditional transactions while capturing the implicit schema-level knowledge encoded in the ontology. Then, traditional association rules algorithms can be applied. We believe this type of learning will become increasingly important in future research both from the machine learning as well as from the Semantic Web communities. Initial experiments on real world Semantic Web data enjoy promising results and show the usefulness of our approach. As future work, we would like to apply generalized query patterns by using the ontology axioms, as well as to automatically discover interesting contexts and their association rules.

References

1. Buitelaar, P., Cimiano, P., Magnini, B. (eds.): *Ontology Learning from Text: Methods, Evaluation and Applications*. *Frontiers in Artificial Intelligence and Applications*, vol. 123. IOS Press, Amsterdam (2005)
2. Muggleton, S., Raedt, L.D.: *Inductive logic programming: Theory and methods*. *J. Log. Program* 19/20, 629–679 (1994)
3. Lisi, F.A., Esposito, F.: *Mining the Semantic Web: A logic-based methodology*. In: Hacid, M.-S., Murray, N.V., Raš, Z.W., Tsumoto, S. (eds.) *ISMIS 2005*. LNCS (LNAI), vol. 3488, pp. 102–111. Springer, Heidelberg (2005)
4. Hartmann, J., Sure, Y.: *A knowledge discovery workbench for the Semantic Web*. In: *Workshop on Mining for and from the Semantic Web at the ACM SIGKDD (August 2004)*
5. Bloehdorn, S., Sure, Y.: *Kernel methods for mining instance data in ontologies*. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) *ASWC 2007 and ISWC 2007*. LNCS, vol. 4825, pp. 58–71. Springer, Heidelberg (2007)
6. Dánger, R., Ruiz-Shulcloper, J., Llavori, R.B.: *Objectminer: A new approach for mining complex objects*. In: *ICEIS (2)*, pp. 42–47 (2004)
7. Rodríguez-González, A.Y., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Ruiz-Shulcloper, J.: *Mining frequent similar patterns on mixed data*. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) *CIARP 2008*. LNCS, vol. 5197, pp. 136–144. Springer, Heidelberg (2008)
8. Chi, Y., Muntz, R.R., Nijssen, S., Kok, J.N.: *Frequent subtree mining - an overview*. *Fundam. Inform.* 66(1-2), 161–198 (2005)
9. Kuramochi, M., Karypis, G.: *Frequent subgraph discovery*. In: Cercone, N., Lin, T.Y., Wu, X. (eds.) *ICDM*, pp. 313–320. IEEE Computer Society, Los Alamitos (2001)
10. Kiefer, C., Bernstein, A., Locher, A.: *Adding data mining support to SPARQL via statistical relational learning methods*. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) *ESWC 2008*. LNCS, vol. 5021, pp. 478–492. Springer, Heidelberg (2008)
11. Kochut, K., Janik, M.: *SPARQLer: Extended SPARQL for semantic association discovery*. In: Franconi, E., Kifer, M., May, W. (eds.) *ESWC 2007*. LNCS, vol. 4519, pp. 145–159. Springer, Heidelberg (2007)
12. Agrawal, R., Imielinski, T., Swami, A.N.: *Mining association rules between sets of items in large databases*. In: *SIGMOD Conference*, pp. 207–216. ACM Press, New York (1993)
13. Nebot, V., Llavori, R.B.: *Efficient retrieval of ontology fragments using an interval labeling scheme*. *Inf. Sci.* 179(24), 4151–4173 (2009)
14. Agrawal, R., Srikant, R.: *Fast algorithms for mining association rules in large databases*. In: *VLDB*, pp. 487–499. Morgan Kaufmann, San Francisco (1994)

Hierarchical Topic-Based Communities Construction for Authors in a Literature Database*

Chien-Liang Wu and Jia-Ling Koh

Department of Information Science and Computer Engineering,
National Taiwan Normal University, Taipei, Taiwan, R.O.C.
wucl@ice.ntnu.edu.tw, jlkoh@csie.ntnu.edu.tw

Abstract. In this paper, given a set of research papers with only title and author information, a mining strategy is proposed to discover and organize the communities of authors according to both the co-author relationships and research topics of their published papers. The proposed method applies the CONGA algorithm to discover collaborative communities from the network constructed from the co-author relationship. To further group the collaborative communities of authors according to research interests, the CiteSeer^X is used as an external source to discover the hidden hierarchical relationships among the topics covered by the papers. In order to evaluate whether the constructed topic-based collaborative community is semantically meaningful, the first part of evaluation is to measure the consistency between the terms appearing in the published papers of a topic-based collaborative community and the terms in the documents related to the specific topic retrieved from other external source. The experimental results show that 81.61% of the topic-based collaborative communities satisfy the consistency requirement. On the other hand, the accuracy of the discovered sub-concept relationship is verified by checking the Wikipedia categories. It is shown that 75.96% of the sub-concept terms are properly assigned in the concept hierarchy.

Keywords: Social Network, Community Mining, Bibliographic database.

1 Introduction

In the area of social network analysis, one important issue is community discovery. A community in a social network is usually defined to be a densely connected sub-graph in the network. By detecting communities, it helps us to understand and exploit the networks more effectively. Especially in a bibliographic database, identifying communities from a co-authorship network can reveal academic activities as well as evolution of research areas; discovering communities in citation network can demonstrate the information diffusion within and between different research areas.

There have been several community mining algorithms proposed to identify meaningful communities from networks. These algorithms can be broadly classified into two main categories: graph partitioning based approaches [2, 12, 13] and modularity

* This work was partially supported by the R.O.C. N.S.C. under Contract No. 98-2221-E-003-017 and NSC 98-2631-S-003-002.

based approaches [3, 4, 11, 15]. Identifying the communities within a network has become one of the major concerns of social network analysis which has various applications. In this paper, we are interested to discover topic-based collaborative communities from co-authorship network.

Several studies have been proposed to analyze bibliographic databases. Zhang et al. [15] proposed the SSN-LDA model to discover flat communities from social networks by utilizing the topological information in social networks. Deng et al. [1] proposed a novel graph-based re-ranking model to improve the ranking of the retrieved documents with respect to a given query. Then the model was used to discover experts of a specific topic from the DBLP bibliographic data. Zaiane et al. [14] provided a new random walk approach to discover research communities with potentially collaborative relationship from the DBLP database [9]. An extended bipartite graph is built to model the relationships of authors and conferences. In order to include the topic information, the proposed model is further extended to be a tripartite graph. Then the random walk with restart algorithm was revised to calculate the relevance scores among researchers in the graph to group the highly-relevant researchers into the same community. Mei et al. [10] proposed the NetPLSA model which combined the statistical topic modeling and social network analysis to discover topical communities. The PLSA model proposed in [6] was exploited to get the weights of the predefined topics for each author. Thus, the topic similarity between each pair of researchers can be evaluated by comparing their topic weights. Moreover, the Harmonic function is used to evaluate the degree of collaborative relationship among each pair of researchers. Accordingly, an objective function is defined by integrating these two models. The topical communities are then discovered by minimizing the objective function.

The previous works [14] and [10] mentioned above are closely related to our work. In [14], for the members in a community discovered from the extended bipartite graph of author-conference relationship, they may have high weighted co-author relationship or often publish papers in the same set of conferences. However, the topics covered in each community are not explicitly specified. Moreover, although the members in a community discovered from the tripartite graph model have similar research topics, it is not necessary that they have strong co-work relationships. Likewise, although [10] regularize a statistical topic model with a harmonic regularization based on a graph structure, the concept hierarchy of topics covered in the communities is not shown explicitly.

In this paper, a mining strategy is proposed for discovering topic-based collaborative community. First, CONGA algorithm [3] is applied to discover overlapping collaborative communities in the collaborative network. By applying the hidden information in the external source CiteSeer^X, the collaborative communities are further organized in a semantic level by automatically constructing a concept hierarchy of topic terms. Therefore, the collaborative communities corresponding to a research topic at arbitrary semantic level can be retrieved. In order to evaluate whether the constructed topic-based collaborative community is semantically meaningful, the first part of evaluation is to measure the consistency between the terms appearing in the published papers of a topic-based collaborative community and the terms in the documents related to the specific topic retrieved from other external source. The experimental results show that 81.61% of the topic-based collaborative communities satisfy the consistency requirement. On the other hand, the accuracy of the discovered sub-concept relationship is verified by checking the Wikipedia categories. It is shown that 75.96% of the sub-concept terms are properly assigned in the concept hierarchy.

The remaining sections of this paper are organized as follows. The related works are discussed in Section 2. Section 3 describes the proposed strategies for discovering the topic-based collaborative communities. The performance study is presented and discussed in Section 4. Finally, Section 5 provides the conclusion and future works.

2 Related Works

Community Discovery. The community mining algorithms can be broadly classified into two main categories: graph partitioning based and modularity based approaches. The partitioning approaches divide the vertices into different communities by minimizing the number of edges between the vertices in different communities, such as the min-max cut algorithm [2], normalized cuts algorithm [12], and spectral clustering algorithm [13]. On the other hand, the modularity based approaches provided a modularity measure to perform good data partitioning during the mining process. One of the representative methods is the Newman algorithm proposed by Newman et al. [11].

The limitation of the Newman algorithm is that it does not allow a node being assigned to more than one community. In reality, an individual may exist in more than one community to take on various roles, such as a blog user being a professional cook and an amateur photographer at the same time. For this reason, Gregory et al. [3] modified the Newman algorithm and proposed the CONGA algorithm, which introduced an operation for splitting a vertex. Suppose a node v is split into v_1 and v_2 , a virtual edge is constructed to connect v_1 and v_2 . The betweenness centrality of the virtual edge is called the split betweenness of node v . In each iterative step of the CONGA algorithm, either the edge with the maximum edge betweenness is removed or the node with the maximum split betweenness is split, depending on which one is greater. The CONGA algorithm is an extension of the Newman algorithm. Thus, it suffers from the huge computation cost for recalculating the betweenness of the nodes and edges repeatedly. For solving this problem, an improved version of the CONGA algorithm, which is named the CONGO algorithm, was proposed by the same author in [4]. In order to speed up the processing efficiency, instead of traversing every edge in the network globally, a parameter h is given to limit the search region locally when updating the betweenness after an edge was removed or a node was split.

Document Clustering. Most traditional document clustering methods adopted the “Bag of Words” (BOW) model to represent a document. A document is thus represented by a term vector; and the similarity of two documents is measured according to their term vectors. However, this approach ignored the relationships between important terms that do not co-occur in the documents, such as synonyms.

Recently, there is a growing amount of tasks on how to utilizing external background knowledge (e.g. WordNet and Wikipedia) to enhance document clustering [7, 8, 5]. Hotho et al. [7] used WordNet, a general ontology, to represent each document by a concept vector instead of a word vector. Furthermore, [8] and [5] considered Wikipedia is a more comprehensive resource to provide potential ontology which can be exploited for enriching text representation. Therefore, the mapping strategies were developed in [8] to match text documents to Wikipedia topics and further to Wikipedia categories. Then the text documents are clustered not only based on the similarity metric of document content but also the concept and category information. In [5], a document was

modeled by a graph of terms with semantic links. By providing a semantic relatedness measure of terms according to Wikipedia, the Newman algorithm was performed to discover the communities of terms for extracting key terms in the document.

3 Topic-Based Collaborative Communities Discovery

3.1 Problem Definition

According to a given bibliographic dataset, the information of the co-authorship between researchers is modeled as a graph $G(V, E)$. Each node v_i in V represents a researcher. Besides, an edge $e=(v_i, v_j)$ in E connecting two nodes v_i and v_j if the two corresponding researchers have at least one co-publishing paper in the dataset. The constructed graph G is called a *collaborative network*.

A densely connected subgraph in graph $G(V, E)$ is called a *collaborative community*, whose nodes represent the researchers with strong co-work relationships. However, the collaborative communities only consider the co-author relationship as the basis of grouping researchers. In order to organize the researchers in a semantic level, a better way is to group the collaborative communities according to the research topics covered in the communities. Moreover, the research topics usually form a concept hierarchy as the example shown in Fig. 1. If the collaborative communities are further assigned to the concept hierarchy of research topics, a hierarchy of the *topic-based collaborative communities* can be constructed. As a result, the users can access the members in the same community not only by their co-author relationship but also the similar research interests at different concept level.

In a bibliographic dataset, suppose only the information of author, co-authors, and title is available for each publication, the challenge is how to extract the research topics covered in a collaborative community and construct the concept hierarchy of topics automatically.

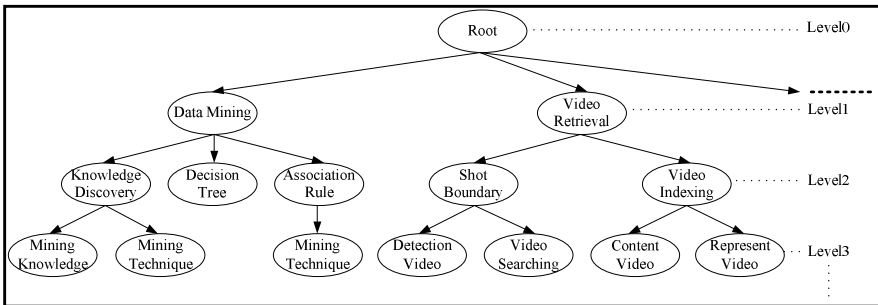


Fig. 1. An example of concept hierarchy of research topics

3.2 Collaborative Community Discovery

We downloaded the bibliographic database from the DBLP website (<http://dblp.uni-trier.de/xml/dblp.xml.gz>). Each data in the DBLP database contains the names of researchers, published paper, journal/conference, year and other related information.

Among the related works of community mining, it is limited that each vertex is assigned to exact one community in most studies. However, in the real world, many researchers have more than one research interest. It is possible that a researcher has ever co-published papers with other researchers in different domains. It is improper to assign the researcher to a collaborative community with specific topic. Therefore, the CONGA algorithm proposed by [3] is used to discover the densely connected sub-graphs with overlapping allowed in the graph. As a result, a researcher with multiple research interests will appear in many collaborative communities.

Fig. 2 shows an example of the discovered collaborative community from a collaborative network, where the nodes labeled by A, B, C, etc. represent the researchers. If two researchers have any co-published paper, a solid edge will connect them. Accordingly, the black dotted-line circles imply the collaborative communities discovered by the CONGA algorithm. It is shown that the researchers A, B, and G all belong to more than one collaborative community.

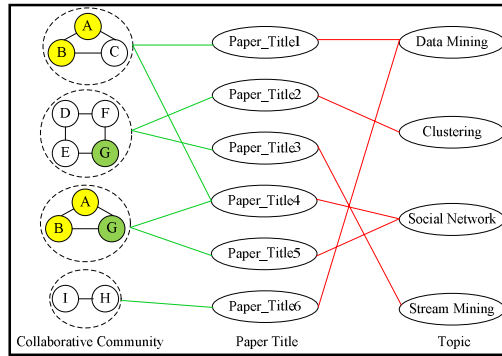


Fig. 2. Collaborative community with corresponding paper topics

3.3 Concept Hierarchy Construction of Research Topics

A topic-based collaborative community is formed by the collaborative communities with a specific topic. The information of the topic covered by a collaborative community is implicit in the corresponding published papers. As Fig. 2 shows, the nodes in the middle are used to denote the titles of papers. A paper title is connected to the collaborative community which contains all its authors. Besides, the rightmost nodes in the Fig. 2 denote the implicit research topics of the papers, such as “Data Mining”, “Clustering”, and “Social Network” etc. If the research topics and the links between paper titles and topics can be extracted automatically, the discovered collaborative communities can be further grouped according the topics covered in their published papers. Using the collaborative communities {A, B, C} and {H, I} shown in Fig. 2 as an example, the authors in these two collaborative communities have never co-published any paper. However, both of the two collaborative communities have a paper with topic “Data Mining”. Therefore, these two collaborative communities should be grouped into a topic-based collaborative community with topic “Data Mining”.

```

Algorithm CCH
Input: all topic terms, Sub_concept(X) for each topic term X
Output: concept hierarchical paths
  For each topic term X
    Call M_DFS (<X>, Sub_concept(X));
Procedure M_DFS (P, Sub_concept)
  For each item t' in Sub_concept
    If (t' is a sub-concept of all the topic terms in P)
      P' = append item t' to P;
      Call M_DFS (P', Sub_concept(t'));
    else output(P);

```

Fig. 3. Pseudo codes for discovering all the concept hierarchical paths

Under the limited information provided in a bibliographic database, only “paper title” best describes the content covered in a paper. Therefore, we perform the following processing to extract the potential topic terms. First, each paper title is processed by the basic text processing steps, including removing stop words and stemming. After that, the bigrams are extracted from the titles. The bigrams with frequency higher than a given threshold α is chosen to be the topic terms.

From the extracted topic terms, it is not easy to determine whether the topics of two papers are related. For example, suppose the title of paper A contains the topic term “Data Mining”, while the title of paper B contains “Sequential Pattern”. Although these two topic terms are different lexically, it is known that “Sequential Pattern” is an important research issue of “Data Mining” in computer science. For solving this problem, the external source CiteSeer^X is used in our approach to construct the hidden concept hierarchy of the extracted topic terms.

For each pair of topic terms X and Y, the confidences of the association rules $X \rightarrow Y$ and $Y \rightarrow X$ are measured to decide whether there exists a hidden sub-concept relationship between X and Y. For getting the confidences of the association rules, each topic term and each pair of topic terms are used as query keywords, respectively, to get the numbers of documents that the two topic terms separate occurrences and co-occurrences in CiteSeer^X. Thus, the $conf(X \rightarrow Y)$ of association rule $X \rightarrow Y$ is obtained from $|D(X \cap Y)|/|D(X)|$ where $|D(X)|$ denotes the number of documents contain topic term X and $|D(X \cap Y)|$ denotes the number of documents contain both X and Y. If $conf(X \rightarrow Y)$ is less than 1 and $conf(Y \rightarrow X)$ is 1, it is implied that if a document contains Y, it must also contain X, but the inverse is not true. In other words, topic term Y is a sub-concept of topic term X. By considering the noise in real data, when deciding whether a topic term Y is a sub-concept of topic term X, the *criterion* is relaxed to require that both $conf(X \rightarrow Y)$ and $conf(Y \rightarrow X)$ are greater than a given threshold value β , and $conf(X \rightarrow Y)$ is smaller than $conf(Y \rightarrow X)$.

The sub-concept relationship among topic terms is then used to construct a concept hierarchy of the topic terms as shown in Fig. 1. The algorithm for discovering all the concept hierarchical paths of the topic terms is described as the pseudo codes shown in Fig. 3. Let $Sub_concept(X)$ denote the set of detected sub-concepts of a topic term

X. If Y is a sub-concept of X, the discovered hierarchy path is denoted as $\langle X, Y \rangle$. If Z is both a sub-concept of X and Y, the constructed hierarchy path is denoted as $\langle X, Y, Z \rangle$. Initially, the $\text{Sub_concept}(X)$ is discovered for each topic term X. The procedure $\text{M_DFS}()$ is called to construct all the concept hierarchical paths existing among the topic terms in a depth-first manner. Let $P = \langle t_1, t_2, \dots, t_n \rangle$ denote a discovered hierarchy path, where $t_i (i=1, \dots, n)$ denote a topic term in the path. A topic term t' can be appended to the path only when t' is a sub-concept of all the topic terms in P .

Since the sub-concept relationship has the transitivity property, only the maximum hierarchical paths have to be maintained. For this reason, the hierarchical paths which are subsequences of any other discovered hierarchical path are removed. Finally, a concept hierarchy of topic terms is then constructed by constructing a prefix tree structure for the discovered hierarchical paths. As shown in Fig. 1, the topic terms located at level 1 represent the most general concepts in the concept hierarchy. The topic terms at level 2 are sub-concepts of their parent. For example, the nodes in the subtree rooted at the node "Data Mining" are all related topic terms in the field of "Data Mining". Besides, the children nodes of the node "Data Mining" represent the sub-concepts in the domain of "Data Mining", such as "Knowledge Discovery", "Decision Tree" and "Association Rule".

Next, according to the established concept hierarchy of topic terms, a collaborative community is assigned to the proper nodes in the concept hierarchy according to its published papers. Let C_i, tt denote the set of topic terms in the titles of the papers whose authors are all in collaborative community C_i . For each topic term t in C_i, tt , if the term t exactly matches to the topic term of a node in the concept hierarchy, C_i is then assigned to this node. Otherwise, the following process is performed to look up the most related topic with t . First, among all the topic terms represented by the nodes at level 1 of the concept hierarchy, the topic t_i with the highest $\text{conf}(t \rightarrow t_i)$ is identified. If $\text{conf}(t \rightarrow t_i)$ is greater than or equal to the given threshold β , the above process will be performed recursively on the nodes in the subtree rooted at the node of topic t_i . The process will continue until the confidence $\text{conf}(t \rightarrow t_i)$ for each topic t_i at the level is smaller than the threshold β . Then the collaborative community C_i is assigned to the parent node of t_i . If the highest $\text{conf}(t \rightarrow t_i)$ obtained at level 1 is less than the given threshold β , the collaborative community C_i is topic-undetermined according to term t . The task of assigning the collaborative community to the concept hierarchy will repeat until all the terms in C_i, tt have been examined.

For a topic t in the concept hierarchy, the corresponding topic-based collaborative community consists of the members in the collaborative communities which are assigned to the subtree rooted at the node of t . Therefore, if the publishing papers of a collaborative community cover multiple topic terms, the collaborative community will belong to multiple topic-based collaborative communities. For each topic-based collaborative community, the number of papers of a member containing the topic term is divided by the total number of papers assigned to the topic term to get the *participating degree* of the member in the community. Moreover, the *concentrate degree* of the member is obtained by dividing the number of papers of the member containing the topic term into his total number of published papers.

4 Experimental Evaluation

4.1 Testing Dataset

In the experimental evaluation, the testing dataset (named 23-CONF) was used, where the papers published from 2006 to 2008 in the 23 data mining related conferences were extracted from the DBLP database.

The corresponding co-authorship network is constructed for the testing dataset, first. After performing the CONGA to discover the collaborative communities in the network, the topic terms are extracted and organized to a concept hierarchy. The related information of the testing dataset is as follows: the number of papers in the dataset is 10800, the number of nodes in the constructed network is 17216, and the number of edges is 34961. The threshold value α for filtering potential topic terms is set to be 0.05, and the threshold value β for detecting the hierarchical relationship between topic terms is set to be 0.13. Finally, the number of discovered collaborative communities is 2230, the number of nodes located at level 1 is 42, the average length of the discovered hierarchical paths is 4.7.

4.2 Evaluation Results

4.2.1 Consistency of a Topic-Based Collaborative Community

In order to evaluate whether the constructed topic-based collaborative community is semantically meaningful, the first part of evaluation is to measure the consistency between the terms appearing in the published papers of a topic-based collaborative community and the terms in other documents related to the specific topic.

An abstract is a concise version of a paper that includes the paper’s research purpose, methodology, experimental results, etc. We thus believe that the abstract of a paper contains more keywords that describe the topic of a paper than the title. Therefore, for each topic term t_i located at level 1 in the discovered concept hierarchy of topic terms, the abstracts of the papers in the corresponding topic-based collaborative community are extracted from CiteSeer^X. There are almost 22% of the papers whose abstracts can be obtained from CiteSeer^X. After performing the text processing steps on the abstracts, the unigrams are extracted from these abstracts to form the set of keywords: B_{t_i} . On the other hand, the ACM (<http://www.acm.org>) digital library is queried to retrieve the abstracts of the most related 200 papers for the topic term t_i . The unigrams extracted from this set of abstracts form another set of keywords: $B_{t_i}^{ACM}$. Then we use the Jensen-Shannon Divergence (*JSD*) to measure the similarity between the probability distributions of terms in two sets of keywords.

Let $\Pr(b|B_{t_i})$ denote the probability of keyword b in B_{t_i} and $\Pr(b|B_{t_j}^{ACM})$ denote the probability of keyword b in $B_{t_j}^{ACM}$. The *JSD* measure between B_{t_i} and $B_{t_j}^{ACM}$ for each pair of topic terms t_i and t_j is shown below.

$$JSD(B_{t_i} \| B_{t_j}^{ACM}) = \frac{1}{2} (KLD(B_{t_i} \| avg(B_{t_i}, B_{t_j}^{ACM})) + KLD(B_{t_j}^{ACM} \| avg(B_{t_i}, B_{t_j}^{ACM}))) \quad (1)$$

$$KLD(B_{t_i} \| avg(B_{t_i}, B_{t_j}^{ACM})) = \sum_{b \in B_{t_i}} \Pr(b|B_{t_i}) \log \frac{\Pr(b|B_{t_i})}{\frac{1}{2}(\Pr(b|B_{t_i}) + \Pr(b|B_{t_j}^{ACM}))} \quad (2)$$

When the probability distributions of terms in the two set of documents are more similar, the *JSD* measure will get lower value. It is indicated that the words appearing in the papers assigned to topic t_i is consistent with the papers searched by topic t_j from the ACM digital library.

In the constructed concept hierarchy of topic terms, there are 42 topic terms located at level 1. For each topic term t_i at level 1, $JSD(B_{t_i}||B_{t_j}^{ACM})$ is measured with all the terms at level 1 one by one. The measuring results are then sorted in ascending order. The corresponding topic t_j of the top 1 result represents the most consistent topic of t_i in the ACM digital library. The evaluation result shows that 81.61% of the topic terms have themselves as their most consistent topics in the ACM digital library. If the top 2 most consistent topics in the ACM digital library are identified, 84.38% of topic terms themselves are covered. By observing the situations that the most consistent topic of a topic term t_i is another topic term t_j , it usually occurs when the topic-terms are cross-domain such as "Neural Network" and "Data Mining". The *JSD* between $B_{\{\text{Neural Network}\}}^{ACM}$ and $B_{\{\text{Data Mining}\}}^{ACM}$ is lower than that between $B_{\{\text{Neural Network}\}}^{ACM}$ and $B_{\{\text{Neural Network}\}}^{ACM}$. The reason is that many important keywords in $B_{\{\text{Neural Network}\}}^{ACM}$, such as "Supervised Learning" and "Markov Model", also appear in $B_{\{\text{Data Mining}\}}^{ACM}$. On the other sides, the papers published in the collaborative communities assigned to "Neural Network" do not contain the popular keywords related to "Neural Network", such as "Gaussian Process" and "Fuzzy Logic". Therefore, the probability distributions of keywords between $B_{\{\text{Neural Network}\}}^{ACM}$ and $B_{\{\text{Data Mining}\}}^{ACM}$ is more similar than that between $B_{\{\text{Neural Network}\}}^{ACM}$ and $B_{\{\text{Neural Network}\}}^{ACM}$.

4.2.2 Accuracy of the Sub-concept Relationship

In this part of evaluation, we would like to measure the correctness of the discovered hierarchical path of the topic terms. Since there is no ground truth to compare with the constructed concept hierarchy, we would like to use the external source Wikipedia to validate the accuracy of our discovered sub-concept relationship.

Let T_level2 denote the topic terms located at equal to or larger than level 2. For each topic term t in T_level2 , it is used as a query term to search on Wikipedia. Let $Category(t)$ denote the set of categories list in Wikipedia for t . If $Category(t)$ contains any super-concept or which is the re-direction of any super-concept in the concept hierarchical path of t , the term t is considered to be properly assigned in the concept hierarchy. The evaluation result shows that 75.96% of the topic terms in T_level2 are properly assigned in the concept hierarchy.

5 Conclusion and Future Works

In this paper, a mining strategy is proposed for discovering collaborative community with similar research interests. The CONGA algorithm is applied to discover overlapping collaborative communities according to the collaborative relationship of authors. In order to organize the collaborative communities at semantic level, the topic terms are extracted from the paper titles, which are automatically constructed into a concept hierarchy by applying the hidden information in the external source CiteSeer^X. Therefore, the resultant topic-based collaborative community provided a semantic-meaningful and flexible view to explore the communities of authors in a bibliographic database. In the experiment, two

evaluation methods are proposed to evaluate the topic consistency of a topic-based collaborative community and accuracy of the discovered sub-concept relationship. The experimental results show that 81.61% of the topic-based collaborative communities satisfy the consistency requirement. Besides, 75.96% of the sub-concept terms are properly assigned in the concept hierarchy.

The evolution analysis of communities and individuals is an interesting issue, which will discover the change of research interests of researchers, the change of the contribution of researchers to a collaborative community, the change of important topic terms. To take the information of publication time into account to detect the dynamic evolution of collaborative communities is under our investigation.

References

1. Deng, H., Lyu, M.R., King, I.: Effective Latent Space Graph-based Re-ranking Model with Global Consistency. In: *Proceeding of the Second ACM International Conference on Web Search and Data Mining*, pp. 212–221 (2009)
2. Ding, C.H.Q., He, X., Zha, H., Gu, M., Simon, H.D.: A Min-max Cut Algorithm for Graph Partitioning and Data Clustering. In: *Proceeding of the IEEE International Conference on Data Mining*, pp. 107–114 (2001)
3. Gregory, S.: An Algorithm to Find Overlapping Community Structure in Networks. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) *PKDD 2007. LNCS (LNAI)*, vol. 4702, pp. 91–102. Springer, Heidelberg (2007)
4. Gregory, S.: A Fast Algorithm to Find Overlapping Communities in Networks. In: *Proceeding of the 12th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 408–423 (2008)
5. Grineva, M.P., Grinev, M.N., Lizorkin, D.: Extracting Key Terms From Noisy and Multi-theme Documents. In: *Proceeding of the 18th ACM International Conference on World Wide Web*, pp. 661–670 (2009)
6. Hofmann, T.: Probabilistic Latent Semantic Indexing. In: *Proceeding of the 22nd ACM SIGIR International Conference on Research and Development in Information Retrieval*, pp. 50–57 (1999)
7. Hotho, A., Staab, S., Stumme, G.: Wordnet Improves Text Document Clustering. In: *Proceeding of the 26th ACM SIGIR International Conference on Semantic Web Workshop*, pp. 541–544 (2003)
8. Hu, X., Zhang, X., Lu, C., Park, E.K., Zhou, X.: Exploiting Wikipedia as External Knowledge for Document Clustering. In: *Proceeding of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 389–396 (2009)
9. Ley, M.: The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. In: *Proceeding of the 9th International Symposium on String Processing and Information Retrieval*, pp. 1–10 (2002)
10. Mei, Q., Cai, D., Zhang, D., Zhai, C.: Topic Modeling with Network Regularization. In: *Proceeding of the 17th ACM International Conference on World Wide Web*, pp. 101–110 (2008)
11. Newman, M.E.J.: Modularity and Community Structure in Networks. *Proceedings of the National Academy of Sciences of the United States of America* 103(23), 8577–8582 (2006)
12. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8), 888–905 (2000)

13. White, S., Smyth, P.: A Spectral Clustering Approach to Finding communities in Graphs. In: Proceeding of the SIAM International Data Mining Conference, pp. 76–84 (2005)
14. Zaiane, O.R., Chen, J., Goebel, R.: DBConnect: Mining Research Community on DBLP Data. In: Proceeding of the First ACM Workshop on Social Network Mining and Analysis, pp. 74–81 (2007)
15. Zhang, H., Qiu, B., Giles, C.L., Foley, H.C., Yen, J.: An LDA-based Community Structure Discovery Approach for Large-Scale Social Networks. In: Proceeding of the IEEE International Conference on Intelligence and Security Informatics, pp. 200–207 (2007)

Generating an Event Arrangement for Understanding News Articles on the Web

Norifumi Hirata, Shun Shiramatsu, Tadachika Ozono, and Toramatsu Shintani

Dept. of Computer Science and Engineering
Graduate School of Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 Japan
{nori,siramatsu,ozono,tora}@toralab.org

Abstract. We propose a new event arrangement system for Web news browsing based on analyzing past news articles related to the browsing-targeted article. Since relevant events are important for understanding news articles, we propose an event arrangement system based on making a connection between the relevant events and providing sequences of those events. When a user chooses an event from candidate events extracted on the basis of time series and important words, the system generates other events related to the chosen one. The system enables a user to find topic sequences suiting one's interest and to closely understand news articles.

1 Introduction

We propose a system that helps a person understand news articles on the web. When we browse a news article on the web, we sometimes find related articles at the end. These related articles help users get the details of the article. The system focuses on related events, not articles. We believe that obtaining related events is important for understanding a news article. The goal of this work is to support understanding of a news article. To understand an article, it is necessary to have background knowledge of the article such as the meaning of words.

The system can detect related events from a news article. The system generates the related events as the event arrangement as follows: first, when the system receives a news article, the system outputs related events in which the system detects events by date and important word. Second, users select the events based on their preferences. Users can get event arrangements by executing these operations.

In Section 2 we explain arranging events of news articles. Section 3 deals with how to arrange events by date and important words. In Section 4 we explain the system structure. In Section 5 we show examples of implementation and discuss the results from experiments. Finally, we comment on our proposed system.

2 Arranging Events of News Articles

There are many related works for analyzing topics for news articles such as Topic Detection and Tracking (TDT) [1,2]. These works usually define a topic or an

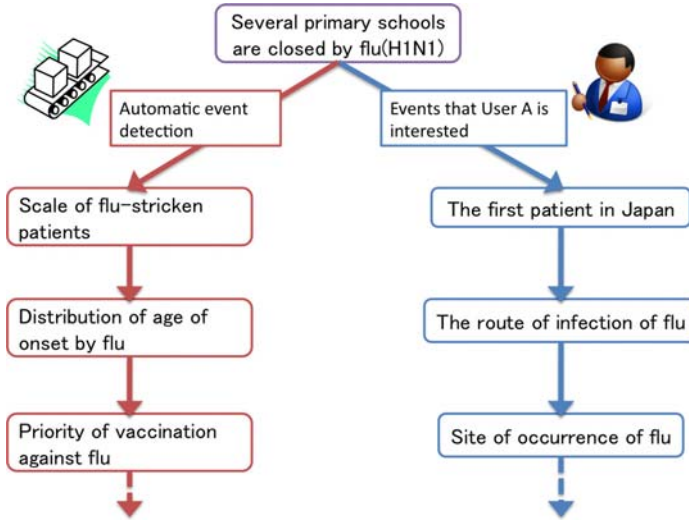


Fig. 1. Arranging events by automatic event detection and user's selection

event, and examine with a corpus of text. In TDT, the notion of a “topic” is modified to be an “event”, meaning a unique occurrence at a point in time. This notion of an event differs from a broader meaning of an event both in spatial/temporal localization and in specificity. However, the notion of topic depends on users.

Arranging events is defined as associating events in this work. In TDT, the notion of a “topic” was modified to be an “event”, meaning a unique occurrence at a point in time. In other research, the notion of an event has a variety of meanings [3,4]. We use the same notion of an event as in TDT. Therefore, an event is composed of news articles. The proposed system detects events, not topics.

Presentation of unrelated events is not helpful for users. Therefore, the system presents events that relate to the news article. Users select an event based on their interests from the presented events. Events are arranged by repeating event presentation and user selection.

The left path in Figure 1 shows an example of an event arrangement that is detected automatically. This example is an event arrangement of school closing by flu. However, an automatic detection method is not sufficient if a user is interested in the route of infection shown in right path in Fig. 1.

When documents are clustered, a method that is often used involves a system collecting documents of a certain period, clustering documents at the point in time, and displaying the result. However, this method must change the system when the system uses new documents and clusters them by user preference. For example, when a system uses new documents, this clusters in each case or uses an applicable method like the leader-follower method [5].

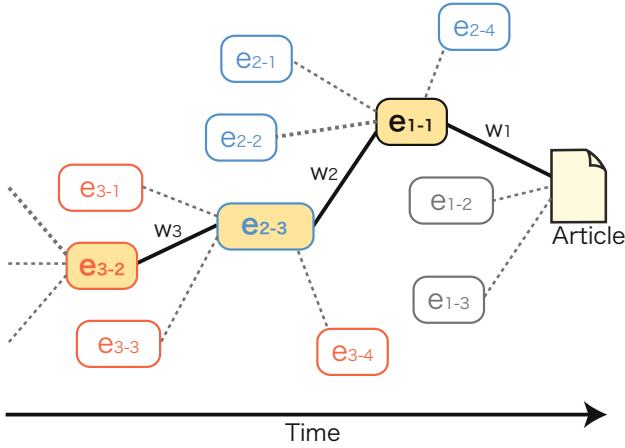


Fig. 2. Arranging events by repeating of event presentation and user selection

Since the system receives users’ requests, it detects events. If the system computes all collected articles or all requests, the computation time is larger. Therefore, the system restricts the computing of articles by retrieval using important words.

Event detection is similar to classification learning. It is well known that preparation of supervised data is costly. Therefore, there have been many studies to reduce the cost. Various methods for reducing the cost have been examined. One of the solutions is to use a method with user feedback. Methods with user feedback can be divided into four types [6,7]. The proposed system uses users’ selection. Therefore, the system is similar to methods with user feedback .

Figure 2 shows how events are arranged by event presentation and user selection. Event arrangement is consisted as a graph. A node is an event, such as e_{i-j} in Fig. 2. If two events are related, the events have an edge. First, when the system receives a new article, it presents related events; e_{1-1} , e_{1-2} , and e_{1-3} . Second, Users select the most interesting event (e_{1-1}) from the presented events. Events related to the selected event are presented such as e_{2-1} , e_{2-2} , e_{2-3} , and e_{2-4} . And, w_1 , w_2 , and w_3 on the edges are important words for detecting each event. By repeating event presentation and user selection, users can receive unique arrangement of events as a graph.

3 Method for Arranging Events

We propose a method of arranging events, repeating four steps; i) important word detection; ii) article retrieval; iii) event detection; iv) event selection. The system retrieves articles using important words, and detects events from retrieved articles. Events are arranged by repeating event detection and user selection.

3.1 Important Word Detection from an Event

Important words are used for event detection. The system filters important words using word classes, since an event is a unique occurrence at a point in time. Important words mean date, location, or what happened. Proper nouns represent locations and actors, and verbs represent actions related to an event. Therefore, important words require proper noun or verb classes. The system excludes some words belong to specific word classes such as a symbol, a particle, an auxiliary verb and a conjunction. It is because they do not represent an event. News articles are delivered as soon as an event occurs. Therefore, we do not use the date on the article as the time an event occurs.

Our system computes the evaluation values of each word, and selects important words that have high-evaluation value. The values are calculated using term frequency - inverse document frequency (tf-idf). When the system receives a class of news articles as an event, it calculates sum of term frequency - inverse document frequency (tf-idf) as shown below:

$$tf \cdot idf(w, e) = \sum_{i=0}^{N-1} tf \cdot idf(w, i) \quad (1)$$

When the system gets an event e which has N articles, it sums up tf-idf values of each article i . If there is no important word w in an article i , $tf \cdot idf(w, i)$ value is zero. In order to calculate the idf, the system counts important words in all the articles that the system has.

3.2 Article Retrieval from Important Words and Date

This subsection explains how to retrieve articles for event detection. Our idea is to use important words for the retrieval. News articles published near the date of the input article are given priority and retrieved. The calculation of the priority is based on a concept that related events occur in about the same time. When the system receives an event, the system processes in the same way. An event date is an average date of news articles that contains the event.

Another method is to compute the similarity of events, which may have higher recall and precision than simple retrieval, but it takes much time. We focus on processing time and our system uses the simple retrieval method for detecting events, though various methods for reducing processing time have been proposed [8,9] and it is difficult to apply them to our system.

3.3 Event Detection

We explain how to detect relevant events from retrieved articles. In a cluster of retrieved articles, our system finds a news article B that is the most similar to the given news article A. The system computes the similarity of two articles using all words including important words in the articles. The similarity in the system is calculated with cosine similarity, and distance function uses group average

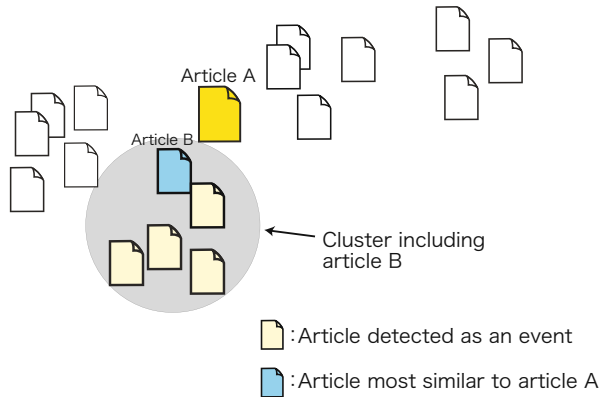


Fig. 3. Event detection from article most similar to input article

method. The system finds a cluster including article B by using a method such as the leader-follower method that is a type of clustering algorithm. The following is details about this method in our proposed system.

- step 1.** Find an article B that is the most similar to the given news article A.
- step 2.** Make a cluster that includes only news article B.
- step 3.** Add a news article to the cluster if the similarity of the news article is higher than the threshold.
- step 4.** Perform step 3 for all retrieved articles.

After this process, the system obtains one cluster that includes news article B, and the cluster is assumed to be a related event. An advantage of this method is that the number of computation is less than the one using the leader-follower algorithm, because our system finds only one cluster.

Figure 3 shows an example of event detection from an given news article A. News articles A and B in Fig. 3 are the most similar in all retrieved articles. The gray circle is a cluster detected as an event.

4 Implementation and Experiment

4.1 Experiment of Event Arrangement System

The structure of our event arrangement system is shown in Figure 4. First, users input a news article into the system, and the system detects important words from the news articles. Second, a news articles that include the important words are retrieved from news-article databases. Finally, the system shows related events from the retrieved results. Therefore, users can obtain events that relate to the browsing news article. Then users select an interesting event from the given events.

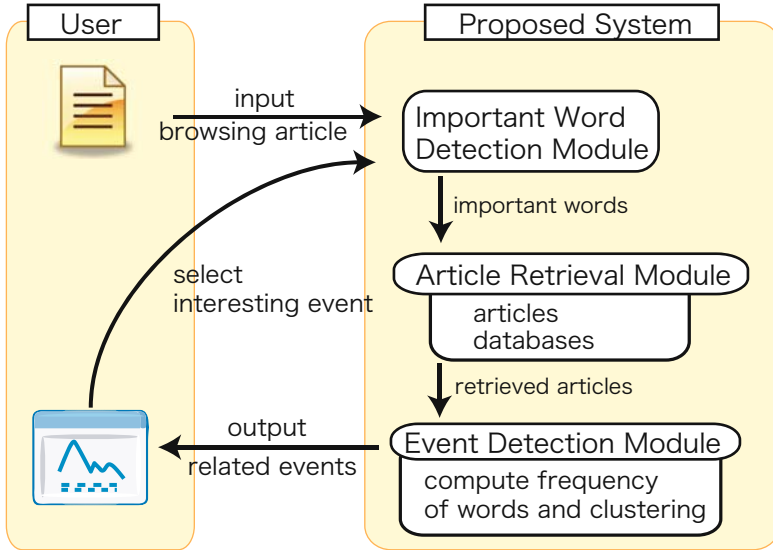


Fig. 4. Structure of event arrangement system

A retrieval module finds news articles excepting the selected events. Therefore, the system does not detect events similar to the presented events.

Dividing a sentence into words is necessary for detection of important words. Japanese is not written with spaces between words; therefore, we use MeCab¹, which is a Japanese language morphological analysis program. After sentences are divided, the system filters important words, and MeCab evaluates word classes.

4.2 Experiment

The purpose of this experiment is to ascertain whether the system is sufficient to obtain an interesting event arrangement for each user. The proposed system uses articles in six news sites. Table 1 lists the six news sites and the number of articles in each site. Period was from November 1st, 2008 to December 31st, 2008. We input a browsing article into the system, and we examine how events were presented.

We show how our system arranges a news event considering user's needs. Before the experiment, the system collects new articles from six Japanese news sites (shown in Table 1). In this experiment, a user wanted to know the occupation of the news titled "Flights have resumed at Bangkok's international airport after anti-government protesters ended their blockade". The original title is Japanese. This English title is the one of "sky news"². This article explains

¹ <http://mecab.sourceforge.net/>

² <http://news.sky.com/skynews>

Table 1. Six news sites and number of articles in each site

| Site name | URL | Number of articles |
|------------------------|-----------------------------|--------------------|
| asahi.com | http://www.asahi.com | 6,688 |
| The Japan Times ONLINE | http://www.japantimes.co.jp | 1,908 |
| Mainichi.jp | http://mainichi.jp | 18,212 |
| NIKKEI NET | http://www.nikkei.co.jp | 21,231 |
| MSN Sankei news | http://sankei.jp.msn.com | 16,105 |
| YOMIURI ONLINE | http://www.yomiuri.co.jp | 3,311 |

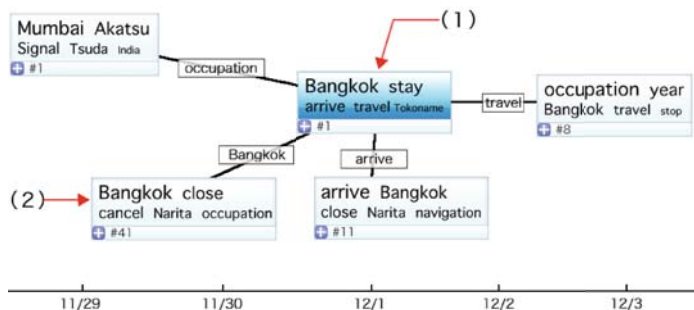


Fig. 5. Events system presented from input article

that “A Thai Airways domestic flight landed at 0715 GMT and a plane bound for Sydney left Suvarnabhumi at 1145 GMT. The occupation of the international and domestic airports left 300,000 tourists stranded in Bangkok for more than a week. ”. Figure 5 shows the result that the system receives the input article. The event in Fig. 5 (1) corresponds to the input article. The system presented four events. Size of the important words in each event was proportional to the evaluation values of the important words. The horizontal axis means when events occur. The presented events can be move by hand. Figure 6 shows a situation in which a user selected the event in Fig. 5 (2). The system presented events related to the selected event as shown in Fig. 6. Then users could obtain event arrangements as shown in Figure 7.

The event arrangement in Fig. 7 was detected by a preferential selection about the past situation in Bangkok. Figure 8 shows another event arrangement. A user could obtain this event arrangement if one selected “flight cancellation” in Fig. 6 and selected “Nagoya”. Presented events changed from events about Bangkok to events about airports and flight.

4.3 Discussion

The system generated an event arrangement as shown in Fig. 7 by repeating event selection. Users can obtain important words such as Bangkok, close, meeting, and support. These important words help users understand the changes in events.

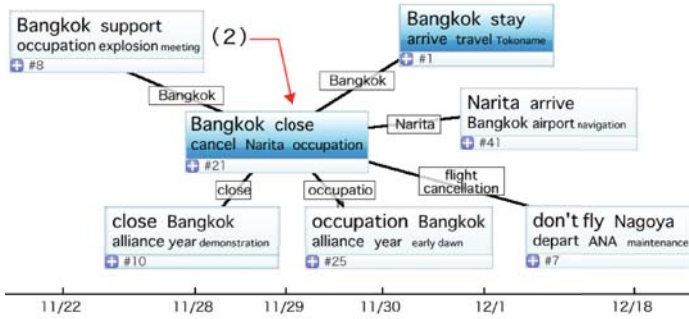


Fig. 6. Events the system presented after selecting event

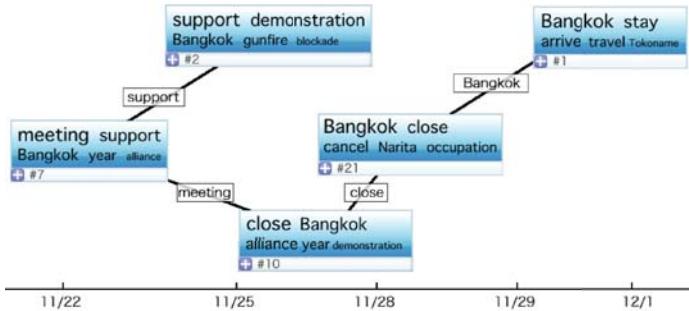


Fig. 7. An event arrangement by repeating event selection

From the results of Fig. 7 and Fig. 8, we confirmed a change of an event arrangement by user selection. Users can get event arrangements along users' interest because the system shows some events and users select an interesting event.

There are two scenarios when the number of articles that are included in an event is small. One is that important words are not sufficient to detect an event. If an important word is a general word, it is difficult to retrieve related articles. The other scenario is that the event is not well known. In this scenario, the number of articles about the event is small. Therefore, the system does not find many relative articles. To solve these problems, evaluation values are needed to improve. For example, it is believed that evaluation values affect the number of retrieved articles.

We examined the effect of word classes. Table 2 lists important words and their evaluation values. The evaluation values in the left table are values using word classes. The right one lists the values not using word classes. "Day" and "service" are not effectual words for detecting events because they are general words. By using word classes, the system can detect important words excluding general words.

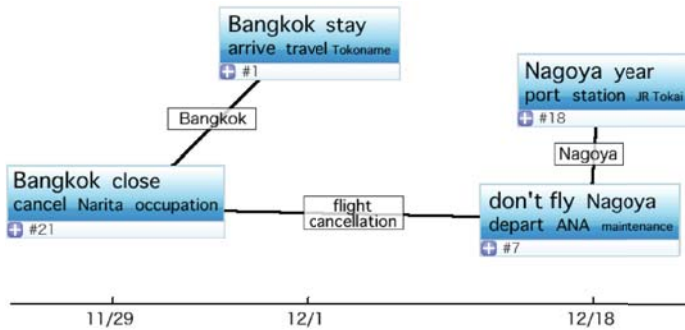


Fig. 8. An event arrangement by another event selection

Table 2. Difference between important words and evaluation values. The title of the input article is “Flights have resumed at Bangkok’s international airport after anti-government protesters ended their blockade”.

| using word class | |
|------------------|------------------|
| important word | evaluation value |
| Bangkok | 0.0652 |
| stay | 0.0455 |
| arrive | 0.0436 |
| travel | 0.0417 |
| Tokoname | 0.0375 |

| not using word class | |
|----------------------|------------------|
| important word | evaluation value |
| airport | 0.1170 |
| service | 0.0864 |
| Bangkok | 0.0652 |
| day | 0.0603 |
| Thailand | 0.0562 |

5 Conclusions

We proposed a system for supporting understanding news articles by arranging events. The system achieves subjective event arrangements by the loop consisting of important word detection step, article retrieval step, and event detection step. Each user can effectively use the event arrangement for understanding news articles. We believe that automatic event detection is not sufficient because users’ interests are different. The event arrangement needs to be generated to each user. In the system, users can arrange events based on their preferences using events recommended by the system. The event arrangement can effectively reflect the interest.

We examined that the system can properly find events by filtering and clustering the articles from articles relevant to selected keywords. In order to track subjective events users need to simply select preferred events from a graph of the events presented by the system. The experimental results using actual news articles show that the proposed system is effective to detect useful events for understanding news articles.

References

1. Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y.: Topic Detection and Tracking Pilot Study Final Report. In: Proceedings of the DARPA broadcast news transcription and understanding workshop, pp. 194–218 (1998)
2. Trieschnigg, D., Kraaij, W.: TNO Hierarchical topic detection report at TDT 2004. In: Topic Detection and Tracking 2004 Workshop (2004)
3. Cieri, C.: Multiple Annotations of Reusable Data Resources: Corpora for Topic Detection and Tracking. In: Proceedings Journées Internationales d'Analyse Statistique des Données Textuelles (2000)
4. Suhara, Y., Toda, H., Sakurai, A.: Extracting Related Named Entities from Blogosphere for Event Mining. In: Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, pp. 225–229 (2008)
5. Duda, R.O., Hart, P.E., Stor, D.G.: Pattern Classification, 2nd edn. John Wiley and Sons, New York (2001)
6. Huang, Y., Mitchell, T.M.: Text Clustering with Extended User Feedback. In: Annual ACM Conference on Research and Development in Information Retrieval, pp. 413–420 (2006)
7. Liu, B., Li, X., Lee, W.S., Yu, P.S.: Text Classification by Labeling Words. In: Proceedings of The Nineteenth National Conference on Artificial Intelligence (2004)
8. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 16–22 (1999)
9. Sahoo, N., Callan, J., Krishnan, R., Duncan, G., Padman, R.: Incremental Hierarchical Clustering of Text Documents. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 357–366 (2006)

Architecture for Automated Search and Negotiation in Affiliation among Community Websites and Blogs

Robin M.E. Swezey, Masato Nakamura, Shun Shiramatsu, Tadachika Ozono,
and Toramatsu Shintani

Nagoya Institute of Technology

Abstract. In this paper, we present a multi-agent architecture which can reduce user's load when searching for affiliates in a network of community websites. We give a precise definition of the environment, networks of community websites. The system's architecture is designed with scalability and easy interfacing for brokers and matchmakers in mind. We have developed a simulator to see how sites or blogs evolve with affiliation. We also show an example of its output results after a 1500-iteration long experiment on a network of community blogs. In our conclusion, we state several applications of critical interest and further research paths on the subject.

1 Introduction

1.1 Aim of the Present Research

Our goal in this research is to provide with a multi-agent architecture capable of automating the process of affiliation in networks of Community Websites (CWs). This process leads to an increase of visitor revenue as well as quality for CWs. Improvement on quality of writing is measured by feedback from visitors, potential affiliates, and page ranking. Increase of the visitor revenue comes from interlinking itself.

Affiliation processes, always part of the process of launching a CW, give birth to a need for automation. Furthermore, new blogs being born everyday make the number of potential affiliates soar. Finding the right affiliate can prove difficult. Affiliation links indeed require an explicit effort compared to that required for permalink ones [1].

In a previous paper [2], we proposed a multi-agent architecture capable of automating this activity, to reduce user's load in time-consuming research and negotiation processes for affiliation. Our system searches for potential affiliate CWs and deals with the issue of equity in partnership, as well as quality expectations, before proposing affiliates to the users. The whole research and negotiation part of the affiliation process becomes automated, thereby saving time for the user. In the case of blogs, it takes the idea of blogs being agents [3] one step further.

In this paper, we deepen the definition of the environment (Sect. 2), and show how our system's simulator can be easily configured and run. We show sample results for a blog community, thereby demonstrating that in this sample case the practical implementation of our system can efficiently reduce user's load in the affiliation process.

1.2 System Description

The practical use case goes as follows:

1. User connects to the broker/provider. If necessary, the user registers the site and information about it, unless the provider is already the blog/site's hosting service.
2. User requests a list of potential affiliates, entering the following data:
 - (a) Desired minimum fairness of the affiliation - in general, or in terms of visitor revenue, quality, relative importance, and so forth.
 - (b) Available spaces to show affiliate links on his own site.
3. System outputs a list of Potential Affiliates (PAs), or none if too unfair.
4. User requests affiliation to one or more PAs and wait for their approval.

The partner PAs need not worry about negotiating, or the user's site being irrelevant to their own expectations. In very simple terms, this use case resembles that of a social networking site, but for community websites.

Our system is also a simulator which generates a cluster of agents, and sample sites based on patterns of particular statistics or expectations. It can be run for any number of iterations. The initial data, as well as the heuristics, depends on the environment chosen. When the simulation ends, the system shows the general evolution of the sites in terms of visitor revenue, quality revenue, and other data if needed.

2 Definitions

2.1 Community Website

A CW is defined by the following characteristics:

1. Its contents are relevant to one particular subject, or several subject of the same category. Ex: one programming language, or mainstream IT. If they are personal sites, they express only one facet of the webmaster [4].
2. It aims for quality of opinions and utterance of relevant and specialist information.

Sites such as daily-life blogs are therefore excluded by this definition. However, blogs of sociological type III [4]), defined as community blogs, are CWs according to our definition.

CWs are set up on the Internet in order to share knowledge and opinions, but before altruism, one of the main objectives is attention in the community [5] (it does not necessarily relates to ego). Therefore, most of the time setting up a community site or blog calls for the process of finding affiliates. Ranking and being well-referenced on search engines calls for quality.

2.2 Affiliation

Affiliation consists in interlinking two websites. For blogs, it is a contract passed between the two, a social tie omnipresent in nowadays' blogs. By placing a link to another weblog in one's blogroll, one assumes that the author either endorses that weblog, wishes to promote it, or claims to read it on a regular basis [1]. This has also been true ever since websites existed on the Internet. Besides friendship or common interest, affiliation seeks to share visitors, as well as raising site awareness mutually. What we define further in this paper as a category is close in concept to an affiliation group [6].

Respective placement (Sect. 4.3) of each other's link on the blogs/websites is considered as settlement of the affiliation deal.

2.3 Visitor Revenue

We call Visitor Revenue (VR) the revenue in popularity that two websites engaging in affiliation seek to increase. It is a general variable that can be associated to hits, pageviews, unique visitors, or real human visitor revenue. In the two latter cases, defining the resource becomes more complex as it may require to define a *reader's behavior*, possibly with reader agents.

2.4 Quality

Quality is the other objective of CWs. It defines the relevance and richness of information itself. Quality can be a ranking based on human classification, usage information, connectivity, or non-affiliated experts [7]. To simplify this notion we consider quality being equivalent to a page ranking, be it in the system itself by different users, or a public page rank. In the experiment featured in the present paper, quality is emulated as a logarithmic function of the visitor revenue, as it can be done approximately for search engines' public page rankings.

3 Society of Community Websites

3.1 Matchmaking and Brokering Architecture

We define a Matchmaking and Brokering Architecture (MBA) (see Fig. 1) fairly similar to that of [8], with a Requester, a Broker (or Server), and a Matchmaker. However, in simulation, we choose not to implement the Requester as a single-threaded agent such as in [9].

It is unrealistic that the Requester, which we describe in our architecture as the Community Website Agent, should possess its own thread running indefinitely on a machine. To simulate intelligent agents and real-time, processing an array of state machines [10] in a loop is sufficient. On each pass, the unit's object is checked for its state and an action course to decide.

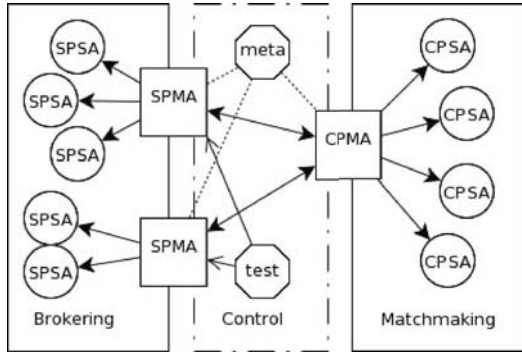


Fig. 1. System Architecture

3.2 Community Website Agent Layer

The CW Agent consists of two parts:

1. The user, holding expectations, and will to request.
2. The knowledge base about the CW. It is stored in a Site Pool Slave Agent (Sect. 3.3).

In practical application, the Requester is indeed a human agent. In simulation, we assimilate them as the same entity: incentives and knowledge are both stored and processed in the Site Pool Layer. Whichever the case, the difference between our architecture and past data mining agent architectures such as [11] is that the data needs not be accessed by external agents which will generate a lot of read accesses and make each request; that part of processing is saved since it is done by the slave database agents themselves (see following section). Since every affiliation request would have to generate a write access either way, to make sure data is up-to-date before sending the request, we use this access to set up a flag for request, and make no unneeded read accesses.

3.3 Brokering: Site Pool Layer

The Site Pool Slave Agents (SPSAs) are agents with a knowledge base about a number of sites. Since every site in each SPSA is independent from the other, it becomes scalable as horizontal sharding. Furthermore, the MiLogPlatform [12] we use for the simulator allows to easily spawn and duplicate agents over networked computers. An infinite loop runs on the SPSAs to simulate virtual Requesters (Sect. 3.1). Once an incentive is set to 1 for one of the sites in the SPSA's array, the site will be part of the next joint request of the current iteration in the SPSA, to the SPMA.

The Site Pool Master Agents (SPMAs) provide control over one or several SPSAs, as well as service or connection to the user's Web interface in practical application. The SPMAs take requests from users, update the incentive state of

the site in the concerned SPSA, then get a joint request from the SPSAs that they will split and submit to the Category Pool Layer (Sect. 3.4) (CPL). They get a response and transmit it to the user. Upon affiliation agreement, they update data again in the SPSAs and submit new placement data (Sect. 4.2) to the CPL. The internal state agents make the final decision and update.

3.4 Matchmaking: Category Pool Layer

The Category Pool Slave Agents (CPSAs) hold data similar to that contained in SPSAs, but the sharding is done by category (Sect. 4.2). They receive requests transmitted by SPMAs to Category Pool Master Agents (CPMAs) from the latter, and are the ones who run the matchmaking algorithm. CPSAs also update their data every N iterations. For example, if an iteration is a day, it is sufficient to update every month. Upon an affiliation agreement, they update the placement data on another request from SPMAs.

The Category Pool Master Agents (CPMAs) are used by the SPMAs to find new affiliation opportunities for the requester site. They control the CPSAs.

3.5 Control Layer

The Meta Agent acts as a directory of all Master Agents in the system so that brokers can find matchmakers.

The Test Coordinator Agent monitors the runs in the simulation and collects data from SPSAs.

3.6 Interface

Agents in the brokering and matchmaking layers share a common inter-agent request interface, for CW and blogging platforms (ex: Blogger, Wordpress) to provide easily with the service to their users. Matchmakers can be independent as well (ex: Blogcatalog). This interface can be seen as a transparent web service.

4 Model

4.1 User Load Reduction Hypothesis

The most common pattern in the affiliation process goes as follows:

1. Incentive of looking for a Potential Affiliate (PA)
2. Actual search for a PA
 - (a) Decide for a tool: search engine, directory, contacts, social networks.
 - (b) Find PA with related topics.
 - (c) Judge the quality of the PA.
 - (d) Match the PA's number of visitors against personal expectation.
3. Negotiations for equity on both sides.
 - (a) Find the right contact for affiliation procedures if there are several.

- (b) Quality evaluation from the PA.
 - (c) Popularity evaluation from the PA.
 - (d) Consider placement of each one’s link on the other’s page.
4. Final agreement.

Whereas only the following steps should be needed, as in our system:

1. Incentive of looking for a Potential Affiliate (PA).
2. Define expected quality and/or visitor revenue from partner.
3. Go through output of the system, make contact immediately.
4. Final agreement.

Therefore, we assume that the system significantly reduces user’s load in the search for affiliates and the following negotiations, if it can succeed in simulation when configured with proper initial data and heuristics about the target environment.

4.2 Knowledge Base

This is the knowledge base contained in every SPSA about each site.

Table 1. Knowledge base

| | |
|------------------------------------------------|-----------------------------------------------------------------------------------------|
| S | The set of all sites in the system. |
| C | The set of all categories. |
| $\forall s \in S, s = (i, c, e, v, q, H_s, a)$ | |
| i | s ’s identifier (URI) |
| $c \in C$ | s ’s category, a set of keywords related to a similar general topic |
| $e \in]-\infty, +\infty[$ | s ’s expectation of fairness |
| $v \in [0, +\infty[$ | s ’s visitor revenue per iteration |
| $q \in [0, 10]$ | s ’s quality rank |
| $H_s / \sum_{h \in H} h \leq 1$ | The placement set |
| $h \in H_s \Rightarrow h \in [0, 1]$ | Value associated to an available placement only |
| $a \in [0, iterations_{run}]$ | Activity rate, an average period in Iterations after which s looks for new affiliates |

4.3 Placement

The space available to put each other’s link on the page, as well as its position and format, is an important matter. There exists no general rule to determine which placement is best on a web page in general: it is influenced by presentation, as well as the number of affiliates already present, and numerous other factors.

For placement of the link, we use a set of probabilities (H) of being accessed from the affiliate, which is independent from the position, format, space, presentation, number of pages the link is to be shown on. In practical application, either the users can fix the values in their H , or this can be done automatically (Sect. 6).

4.4 Matchmaking Algorithm

The matchmaking layer receives all the data about the site from the brokering layer. The CPMA dispatches the requester site's data to the appropriate CPSAs' queues, merge the result arrays and send response to the SPMA. We name the utility function u .

```

dealss ← empty array
for all  $s' \in \text{CPSA}$  do
  dealss,s' ← empty array
  utilitys' ← 0
  for all  $h' \in H'$  do
    for all  $h \in H$  do
       $u \leftarrow h' \times v' \times q' - h \times v \times q$ 
      if  $e \leq u \leq -e'$  then
        INSERT(dealss,s', [ $s', u, h', h$ ])
        utilitys' ← utilitys' +  $u$ 
      end if
    end for
  end for
  if dealss,s' ≠ empty array then
    INSERT(dealss, [dealss,s', utilitys'])
  end if
end for
SORT(dealss, utilitys, desc)
RETURNdealss

```

Fig. 2. Matchmaking algorithm

5 Simulation and Sample Results

In this simulation we show the evolution of a young blog community network.

5.1 Initial Data and Heuristics

Since the goal of our system is to reduce user's load, we expect at least a behavior similar to that of a real blog community: when compared to a network where no affiliation occurs, overall inequality in visitor revenue as well as quality (here, a page ranking, simplified model and function of v), should shrink. Overall visitor revenue should increase.

5.2 Results

We made two runs of the system, one without affiliation process at all (first run, Fig. 3 and 4), and one with each blog requesting periodically for an affiliation (second run, Fig. 5 and 6). Each graph shows VR per Iteration (VRpI) of a blog

Table 2. Experiment parameters

| | |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Iterations | 1500 |
| S | 500 blogs |
| C | 20 categories, affected randomly |
| e | 0 (mean), 100 (deviation) (signed Gaussian generation) |
| $v_{t=0}$ | 100 (mean), 1000 (deviation) (unsigned Gaussian generation) |
| q | Logarithmic function of v , interpolated from real page ranks: $q(0) = 0, q(500) = 1, q(1000) = 2, q(3000) = 3, \dots, q(729000) = 8, \dots$ |
| H | Available slots in blogroll placed at half-height, being placed on a lower slot decreases exponentially click probability |
| a | Random variable: average of one request per blog in 10 iterations |
| Agents | 1 meta agent, 1 SPMA, 8 SPSA, 1 CPMA, 8 CPSA, 1 test coordinator |

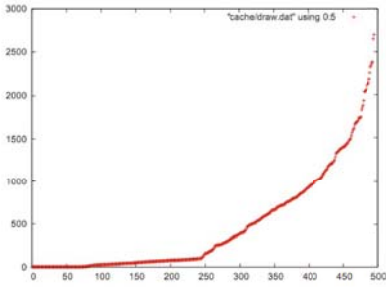


Fig. 3. No Affiliation, $t=1$

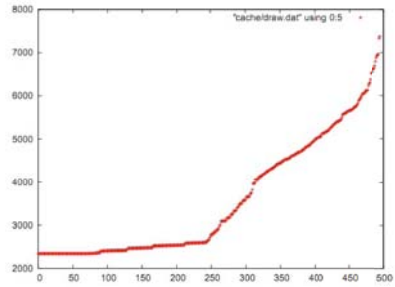


Fig. 4. No Affiliation, $t=1500$

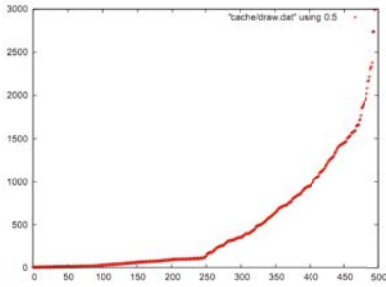


Fig. 5. Affiliation, $t=1$

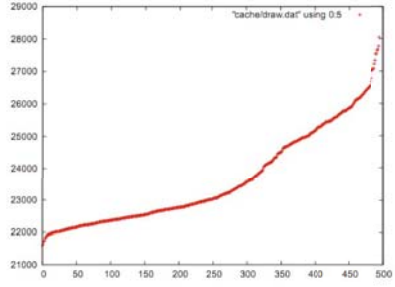


Fig. 6. Affiliation, $t=1500$

in function of the blog id (500 ids). The ids are ordered ascendantly in function of the VRpI so that the graph can be easily read.

In the first run, blogs evolve independently and top quality blogs (right edge of the curve) reach almost 400% of the VRpI of non-popular blogs (left edge of the curve) with a VRpI of 8000 to 2500, with no exchanged visitors. Since there is no cooperation and the rich get richer, the shape of the curve is a long tail. In the second run, blogs help each others. This time, the VRpI of top blogs (right

edge of the curve) is only 132% of the VRpI of least popular blogs, with a VRpI of 28000 to 22000. We can see, as expected, a huge overall increase in VR, thanks to the exchange of visitors coming from affiliation between the blogs. Therefore, the system meets our expectations in this sample experiment, it has managed in reducing VR inequality in a community of blogs.

What is interesting is also that we have verified in simulation the results of the real-world BlogDex study [1], since the shape of our curve remains a long tail. Even if overall inequality has decreased, the rich still tend on getting richer. Encouraging change of this behavior will be the subject of a further paper.

6 Conclusion

In this paper, we have developed an architecture for a system capable of relieving the user of the load of searching and negotiating in the process of affiliation in a network of community websites. Our simulator, of which we have given sample results, makes us able to easily see the evolution of the blogs/sites in a network. Using such a system, users become able to target easily their affiliates and focus more on their writing and contents, and increase the quality of their sites.

We consider adding fair counterparts to affiliation contracts, for sites with other advantages than visitor revenue and quality. Also, other heuristics than public page ranks, such as the Eigen Rumor Algorithm [13], can be input in the simulator for our system. We also intend on testing other patterns of communities after harvesting more data. We will discuss the subject of quality, influence and authority more precisely in a further paper.

In another paper [14], we have been developing a system capable of fetching detailed statistics about the real visitor revenue, the click probabilities (see H in Sect. 4.3), and push advertisement links automatically. We consider plugging it on the present system, for affiliation. This will also be the subject of a paper to come. Moreover, as the use of trackback and similar tools broadens on the Internet, it may prove useful to extend the system to the research and negotiation for trackbacks and references on article pages.

Finally, as of now, the system is not capable of finding sites which are exterior to it. This feature can be developed by setting up brokers/providers that will, instead of requiring registration, crawl the Web. As well, for keywords and categories, recent advances in relational learning [15] could be applied to communities, in order to sharpen the different fields. We are considering this research as well.

References

1. Marlow, C.: Audience, structure and authority in the weblog community. In: International Communication Association Conference, New Orleans, LA, Citeseer (May 2004)
2. Swezey, R., Nakamura, M., Shiramatsu, S., Ozono, T., Shintani, T.: Intelligent and Cooperative Blog Communities. In: Proceedings of the 8th Forum on Information Technology, IPSJ, p. 2 (2009)

3. Takeda, H.: Evolution of the Web, Agents, and Semantic Web. *IPSJ Magazine* 48(3) (March 2007)
4. Cardon, D., Delaunay-Teterel, H., Cédric, F., Prieur, C.: Sociological Typology of Personal Blogs. In: *International Conference on Weblogs and Social Media*, Boulder, Colorado, USA (2007), <http://www.icwsm.org/papers/paper43.html> (accessed on April 27, 2007)
5. Dessalles, J.: Altruism, status and the origin of relevance. *Approaches to the Evolution of Language*, 130–147 (1998)
6. Zheleva, E., Sharara, H., Getoor, L.: Co-evolution of social and affiliation networks. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1007–1016. ACM, New York (2009)
7. Bharat, K., Mihaila, G.: When experts agree: using non-affiliated experts to rank popular topics. *ACM Transactions on Information Systems (TOIS)* 20(1), 47–58 (2002)
8. Decker, K., Williamson, M., Sycara, K.: Matchmaking and brokering. In: *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS 1996)*, p. 432 (1996)
9. Woolridge, M., Jennings, N.: *Intelligent Agents: Theory and Practice*. Cambridge University Press, Cambridge (1995)
10. Woolridge, M.: *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., USA (2001)
11. Kargupta, H., Hamzaoglu, I., Stafford, B.: Scalable, distributed data mining using an agent based architecture. In: *Proceedings the Third International Conference on the Knowledge Discovery and Data Mining*, pp. 211–214. AAAI Press, Menlo Park (1997)
12. Fukuta, N., Ito, T., Shintani, T.: A logic-based framework for mobile intelligent information agents. In: *The Proc. of WWW10*, Citeseer, pp. 58–59 (2001)
13. Fujimura, K., Tanimoto, N.: Ranking Weblogs by Eigen Rumor Algorithm. *Shakai Joho Shisutemugaku Shinpojiiumu Gakujutsu Koen Ronbunshu* 11, 67–72 (2005)
14. Nakamura, M., Asami, S., Ozono, T., Shintani, T.: A Dynamic Rearrangement Mechanism of Web Page Layouts Using Web Agents. In: *Proceedings of the 22nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems: Next-Generation Applied Intelligence*, p. 643. Springer, Heidelberg (2009)
15. Tang, L., Liu, H.: Relational learning via latent social dimensions. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 817–826. ACM, New York (2009)

Effect of Semantic Differences in WordNet-Based Similarity Measures

Raúl Ernesto Menéndez-Mora^{1,2} and Ryutaro Ichise¹

¹ National Institute of Informatics

2-1-2 Hitotsubashi Chiyoda-ku, Tokyo, 101-8430 Japan

² Facultad de Informática y Matemática, Universidad de Holguín

Ave. XX Aniversario, Piedra Blanca, Holguín, 80100 Cuba

{menendez,ichise}@nii.ac.jp

Abstract. Assessing the semantic similarity of words is a generic problem in many research fields such as artificial intelligence, biomedicine, linguistics, cognitive science and psychology. The difficulty of this task lies in how to find an effective way to simulate the process of human judgement of word similarity. In this paper, we introduce the idea of semantic differences and commonalities between words to the similarity computation process. Five new semantic similarity metrics are obtained after applying this scheme to traditional WordNet-based measures. In an experimental evaluation of our approach on a standard 28 word pairs dataset, three of the measures outperformed their classical version, while the other two performed as well as their unmodified counterparts.

Keywords: WordNet Measures, Semantic Similarity, Featured Based Similarity.

1 Introduction

The semantic similarity of words has become a topic of many research fields such as artificial intelligence, biomedicine, linguistics, cognitive science, and psychology. Essential for human categorization and reasoning, semantic similarity is extensively used in a variety of applications, like words sense disambiguation [14], detection and correction of malapropisms [1], information retrieval [4], automatic hypertext linking and natural language processing. Several applications to the field of artificial intelligence are discussed in [16]. However, despite numerous practical applications today, its theoretical foundations lie elsewhere, in cognitive science and psychology where it was the subject of many investigations and theories (e.g., [17]).

Let take a current example of peer-to-peer networks [3] into which semantic similarity has found its way. Assuming a shared taxonomy among the peers to which they can annotate their content, similarities among peers can be inferred by computing similarities among their representative concepts in the shared taxonomy. In this way, the more two peers are similar, the more efficient it is to

route messages toward them. Numerous similar applications are the reasons for the increasing interest in this subject, whose ultimate goal is to mimic human judgement regarding similarity of word pairs.

Semantic similarity of words is often represented by the similarity between the concepts associated with the words. Several methods have been developed to compute word similarity, mostly those operating on the taxonomic dictionary WordNet [2] and exploiting its hierarchical structure. But the majority of them suffer from a serious limitation; they only focus on the semantic information shared by those concepts, i.e., on the common points in the concept definitions. The increasing need for better measures and the new study area of semantic differences between words has led us to this study in the hope of upgrading existing semantic similarities. In particular, we combined traditional WordNet based semantic similarity measures with the idea of the “similarity between entities being related to their commonalities as well as to their differences”, in order to improve the performance of WordNet based similarity measures and to obtain better results for applications using semantic similarities.

The paper is structured as follows. The next section reviews some background knowledge and related work. Section 3 describes our model, as well as the modified metrics. Section 4 discusses the results of an experiment, and section 5 summarizes our work, draws some conclusions, and outlines future work.

2 Related Work

2.1 Semantic Similarity

The key to calculating semantic similarity lies in simulating human thinking behavior. Semantic similarity of words is determined by processing first-hand information sources in the human brain. Semantic similarity is a concept whereby a set of documents or terms within term lists are assigned a metric based on the likeness of their meaning / semantic content.

Some studies have tried to assess the semantic proximity of two given concepts in order to improve the semantic similarity computation. These studies focus on similarity and they use synonymy¹, hyponymy² [19], meronymy³ and other arbitrarily typed semantic relationships. These relationships can be used to connect concepts in graph structures. They are the key ideas behind measures developed to assess the semantic similarity of concepts, i.e., how much one concept has to do with a different one. However, the measures tend to focus on the common points in the concepts’ definitions; they rarely consider semantic differences, and this leaves a big gap in the semantic similarity computation process.

2.2 WordNet

A number of semantic similarity computation methods operate on the taxonomic dictionary WordNet and exploit its hierarchical structure. WordNet [2] is

¹ A semantic relation that holds between two words that can (in a given context) express the same meaning.

² The semantic relation of being subordinate or belonging to a lower rank or class.

³ The semantic relation that holds between a part and the whole.

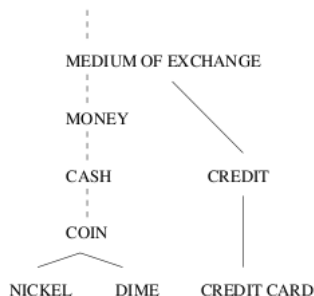


Fig. 1. Fragment of the WordNet taxonomy. Solid lines represent IS-A links; dashed lines indicate that some intervening nodes were omitted to save space.

a machine-readable lexical database that is organized by meanings, and it was developed at Princeton University. Synonymy, hyponymy, meronymy and many other relationships between concepts are represented in this lexical network of English words. WordNet, as an ontology, is intended to model the human lexicon, and psycholinguistic findings were taken into account during its design. It is classified as a light-weight ontology, because it is heavily grounded on its taxonomic structure employing the IS-A inheritance relation, and as a lexical ontology, because it contains both linguistic and ontological information [9]. Figure 1 taken from [13] shows a fragment of WordNet's structure.

Nouns, verbs, adjectives, and adverbs are each organized into networks of synonym sets (*synsets*) each representing one underlying lexical concept and are interlinked with a variety of relations. A polysemous⁴ word will appear in one synset for each of its senses. The backbone of the noun network is the subsumption hierarchy (*hyponymy/hypernymy*), which accounts for close to 80% of the relations in WordNet.

2.3 Semantic Similarity Measures and WordNet

Based on WordNet and depending on the elements taken into consideration, semantic similarity measures can be classified into two different types: *edge-based* and *node-based semantic similarity measures*.

An intuitive way to quickly compute the semantic similarity between two nodes of a hierarchy is to count the number of edges in the shortest path between these two nodes. The idea behind this is that the semantic distance of two concepts is correlated with the length of the shortest path to join these concepts. This measure was first defined by Rada in [12]. However, it relies upon the assumption that each edge carries the same amount of information, which is not true in most ontologies [13]. Many other formulas have since extended Rada's measure by computing weights on edges by using additional information, such

⁴ Polysemy: The ambiguity of an individual word or phrase that can be used (in different contexts) to express two or more different meanings.

as the depth of each concept in the hierarchy and the *lowest common superset, or subsumer (lcs)* [18]. For example, in Figure 1, the *lcs* between the concepts *nickel* and *dime* is the concept *coin*.

The measures which focus on structural semantic information (i.e., the depth of the lowest common superset ($lcs(c_1, c_2)$), the depth of the concept's nodes, and the shortest path between them) are called *edge-based similarity measures*. The Wu & Palmer [18] and Leacock & Chodorow [6] similarity measures are bases in a linear model, whereas Li et al.'s approach [7] combines structural semantic information in a nonlinear model. Li et al.'s model empirically defines a similarity measure that uses the shortest path length, depth, and local density in a taxonomy. They include two parameters which represent the contribution of the shortest path length and the depth of the *lcs* in the similarity computation process.

Another way to compute the similarity between two nodes is by associating a weight with each node. Such similarity measures are called *node-based similarity measures*. From the perspective of information theory, this weight represents the *information content (IC)* of a concept. IC can be considered to be a measure that quantifies the amount of information a concept expresses. The more specialized a concept is, the heavier its weight will be.

The literature contains two main ways of computing information content. The most classical way is Resnik's approach with a corpus [13]

$$IC(c) = -\log p(c) \quad (1)$$

where $p(c)$ is the probability of concept c in the taxonomy. Seco's approach [11] exploits the notion of *intrinsic IC* which quantifies IC values by scrutinizing how concepts are arranged in an ontological structure

$$IC(c) = 1 - \frac{\log(hypo(c) + 1)}{\log(\max_{wn})} \quad (2)$$

where *hypo* returns the total number of hyponyms of a given concept c and \max_{wn} is a constant that indicates the total number of concepts in the corresponding WordNet taxonomy. This definition of IC enables obtaining IC values in a corpus-independent way.

The node-based similarity measures include the metrics of Resnik [13], Jiang & Conrath [5], Lin [8] and Pirró & Seco [11].

In 1977, Tversky presented an abstract model of similarity [17] that takes into account features that are common to two concepts and features specific to each. That is, the similarity of concept c_1 to concept c_2 is a function of the features common to c_1 and c_2 , those in c_1 but not in c_2 and those in c_2 but not in c_1 . Admitting a function $\psi(c)$ that yields the set of features relevant to c , he proposed the following similarity function:

$$Sim_{tvr}(c_1, c_2) = \alpha F(\psi(c_1) \cap \psi(c_2)) - \beta F(\psi(c_1)/\psi(c_2)) - \gamma F(\psi(c_2)/\psi(c_1)) \quad (3)$$

where F is some function that reflects the salience of a set of features, and α , β and γ are parameters provided for differences in each component. According

to Tversky, similarity is not symmetric, that is, $Sim_{tvr}(c_1, c_2) \neq Sim_{tvr}(c_2, c_1)$, because humans tend to focus more on one object than on the other depending on the way the relationship direction is taken into consideration during the comparison. For example, regarding the concept *dime* in Figure 1, it is logical that one of its most related concepts is *nickel*, but the same is not true in the opposite direction. The concept *nickel* is also like *gold*, *metal*, etc.

The Pirró & Seco [11] similarity metric is based on Tversky's theory [17] but from an information-theoretic perspective. This measure achieves very good results in the comparison to human judgments when it is combined with the notion of *intrinsic information content*.

$$Sim_{P\&S}(c_1, c_2) = \begin{cases} 3IC(lcs(c_1, c_2)) - IC(c_1) - IC(c_2) & \text{if } c_1 \neq c_2 \\ 1 & \text{if } c_1 = c_2 \end{cases} \quad (4)$$

Despite all this previous work, WordNet based semantic similarity measures still have problems, which we will discuss in the next section.

3 Menendez-Ichise Model

3.1 Approach

Most of the WordNet based semantic similarity measures just take into consideration semantic commonalities among concepts for computing their values. The strength of semantic differences has been diminished or not exploited at all. Having all these elements in mind and considering the current structure of WordNet, we propose the Menendez-Ichise model. In this section, we introduce our model and its application to traditional WordNet based similarity metrics. The modifications to those metrics are founded on Tversky's feature-based theory of similarity [17].

Our model supports to be a specialization of Tversky's featured-based theory applied to traditional WordNet based similarity metrics. Paraphrasing Tversky, we state that: "the similarity between two entities is related to their commonalities as well as to their differences", and our general model is described by the following expression:

$$Sim(c_1, c_2) = \alpha * Comm(c_1, c_2) - \beta * Diff(c_1, c_2) \quad (5)$$

where $Comm(c_1, c_2)$ stands for **commonalities**, $Diff(c_1, c_2)$ the **differences**, and α and β tuning factors ($0 \leq \alpha, \beta \leq 1$) that represent the importance of the commonalities and differences in the model. Because WordNet's structure is represented by an undirected graph we can't avoid assuming symmetry where there is none.

The use of semantic differences for computing semantic similarity is a novel approach. In the next section, we explain how we applied our model to WordNet based semantic similarity measures.

3.2 Semantic Differences in WordNet Based Metrics

The main features considered by WordNet based similarity metrics are, the distance between nodes and the weight of the nodes. This in turn leads to two different approaches: *edge-based* and *node-based*, as mentioned above.

In our model, independently of the approach used, we consider the information from the *root*⁵ to the *lcs* to be the **semantic commonalities** of the concepts c_1 and c_2 ; and the rest of the information from the *lcs* to each of the concepts c_1 and c_2 to be the **semantic differences**. Hence, from the perspective of an edge-based approach, the differences are related to the shortest path between the two concepts. In node-based approach, the differences are related to the information contained in the nodes representing the concepts that it is not contained in their *lcs*. For example, regarding the concepts *nickel* and *dime* in Figure 1, the semantic commonalities are in their *lcs*, i.e, the taxonomy subgraph from the *root* to $lcs(nickel, dime) = coin$. The semantic differences between both concepts is enclosed in the taxonomy subgraph from $lcs(nickel, dime)$ to both concepts but without any information from the *root* to *coin*.

Equation 6 is a modification to the traditional length and path metrics where we consider the first term to be the semantic commonalities between the concepts and the second term to be their semantic differences. Previous formulas consider either of these features, but not both.

$$Sim'_{length}(c_1, c_2) = \alpha * \frac{1}{2 * depth(lcs(c_1, c_2))} - \beta * \frac{1}{length(c_1, c_2)} \quad (6)$$

The Wu & Palmer and Leacock & Chodorow measures rely on the length of the shortest path between two synsets. Equations 7 and 8 consider the **semantic differences** to be the distance between these two synsets, which is not taken into consideration in their original formulation, and each case is normalized with a different normalization factor. For Wu & Palmer measure, this means the addition of the concepts' depths in the taxonomy; and for Leacock & Chodorow metric, it means twice the taxonomy depth.

$$Sim'_{wup}(c_1, c_2) = \alpha * \frac{2 * depth(lcs(c_1, c_2))}{depth(c_1) + depth(c_2)} - \beta * \frac{length(c_1, c_2)}{depth(c_1) + depth(c_2)} \quad (7)$$

$$Sim'_{lch}(c_1, c_2) = \alpha * (-\lg(\frac{depth(lcs(c_1, c_2))}{2 * \lambda})) - \beta * (-\lg(\frac{length(c_1, c_2)}{2 * \lambda})) \quad (8)$$

The modified Resnik measure considers the **semantic commonalities** to be the information content of the *lcs* and the **semantic differences** to be the information content encompassed by concepts, minus the one already considered in the *lcs*.

$$Sim'_{res}(c_1, c_2) = \alpha * IC(lcs(c_1, c_2)) - \beta * (IC(c_1) + IC(c_2) - 2 * IC(lcs(c_1, c_2))) \quad (9)$$

⁵ The most abstract node in the taxonomy.

The modified Jiang & Conrath similarity expression $Sim'_{j\&c}(c_1, c_2)$ is identical to the one obtained for Resnik's measure, Equation (9), and it is a generalization of the Pirró and Seco similarity measure, Equation (4).

$$Sim_{P\&S} \subset Sim'_{Res}(c_1, c_2) = Sim'_{j\&c}(c_1, c_2) \quad (10)$$

According to Lin [8] “the similarity between c_1 and c_2 is measured by the ratio between the amount of information needed to state the commonality of c_1 and c_2 and the information needed to fully describe what c_1 and c_2 are”. In Equation (11), we add the **semantic differences** as the information content in each concept minus the one already considered in the *lcs* divided by the information needed to fully describe the concepts.

$$Sim'_{lin}(c_1, c_2) = \alpha * \frac{2 * IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} - \beta * \frac{IC(c_1) + IC(c_2) - 2 * IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} \quad (11)$$

4 Experiments and Results

4.1 Experimental Settings

The purpose of the experiment was to evaluate the new semantic similarity measures and to establish a baseline for comparison of their results with those of the original versions. We used the human judgments of Pirró and Seco experiment [11] (P&S in the following) for the word pairs of the Miller and Charles dataset (M&C in the following).

Unfortunately, there is a distinct lack of standards for evaluating semantic similarities, which means that the accuracy of a computational method for evaluating word similarity can only be established by comparing its results against human common sense. That is, a method that comes close to matching human judgments can be deemed accurate. Moreover, some datasets for this evaluation are commonly used. In particular, the Rubenstein and Goodenough dataset (R&G in the following) and Miller and Charles dataset (M&C) are standards dataset for evaluating semantic similarities.

In 1965, Rubenstein and Goodenough [15] obtained “synonymy judgments” of word pairs by hiring 51 subjects to evaluate 65 pairs of nouns. The subjects were asked to assign a similarity from 0 to 4, from “semantically unrelated” to “highly synonymous”. Miller and Charles [10], 25 years later, extracted 30 pairs of nouns from the R&G dataset and repeated their experiment with 38 subjects. The M&C experiment achieved a correlation of 0.97 with the original experiment of R&G. Resnik [13], in 1995, replicated the M&C experiment with 10 computer science students, obtaining a correlation of 0.96. Pirró and Seco [11] (P&S) in 2008 also recreated the R&G experiment this time with 101 subjects, and arrived at a correlation coefficient of 0.972 for the full dataset ($P\&S_{full}$).

As mentioned above, we used the judgments of the P&S experiment for the word pairs of the M&C dataset. However, we considered only 28 word pairs of

Table 1. Correlation coefficients for the different values of β

| | Original | 0.0 | 0.1 | 0.2 | 0.4 | 0.5 | 0.6 | 0.7 | 0.9 | 1.0 |
|--------|----------|--------|--------|--------|--------|--------|---------------|--------|--------|---------------|
| length | 0.8401 | 0.6673 | 0.7958 | 0.8358 | 0.8550 | 0.8568 | 0.8571 | 0.8568 | 0.8556 | 0.8549 |
| wup | 0.7726 | 0.7726 | 0.7726 | 0.7726 | 0.7726 | 0.7726 | 0.7726 | 0.7726 | 0.7726 | 0.7726 |
| lch | 0.8293 | 0.7126 | 0.7446 | 0.7658 | 0.7907 | 0.7983 | 0.8039 | 0.8083 | 0.8144 | 0.8165 |
| res | 0.8308 | 0.8308 | 0.8433 | 0.8508 | 0.8587 | 0.8609 | 0.8624 | 0.8635 | 0.8650 | 0.8655 |
| lin | 0.8587 | 0.8587 | 0.8587 | 0.8587 | 0.8587 | 0.8587 | 0.8587 | 0.8587 | 0.8587 | 0.8587 |
| j&c | 0.8660 | 0.8308 | 0.8433 | 0.8508 | 0.8587 | 0.8609 | 0.8624 | 0.8635 | 0.8650 | 0.8655 |

the 30 used in the M&C experiment: a word missing in WordNet made it impossible to compute ratings for the other two word pairs. All the evaluations were performed using WordNet 3.0 [2] and the Brown Corpus⁶ was used for the information content based metric calculation. The computation used Pedersen’s WordNet::Similarity Perl module as the core. We also recreated the P&S experiment with the Java WordNet Similarity Library [11] (JWSL) using Pirró and Seco’s *intrinsic information content*, but we did not obtain the same results they did.

For the new metrics we perform two experiments, both varying the importance of the semantic difference’s factor β , and then calculating the correlation with the human judgments values of the P&S experiment. In the first experiment β takes values in the range of [0,1] while in the second experiment $\beta > 1$. The importance of the semantic commonalities factor was kept constant ($\alpha = 1$), since we wanted to focus on the effect of semantic differences in WordNet based measures.

4.2 Results and Discussion

Table 1 compiles the results of the first experiment for several *difference’s factors*. The first value ($\beta = Original$) represents the original measure⁷, i.e., the previous result. The correlation value when $\beta = Original$ and $\beta = 0.0$ should be the same if the modified measure considers the commonalities as in the original metric. This is not the case of $Sim'_{length}(c_1, c_2)$, $Sim'_{lch}(c_1, c_2)$ and $Sim'_{j\&c}(c_1, c_2)$; and it is the reason for the differences in their correlation values when $\beta = Original$ and $\beta = 0.0$.

Table 2 shows the result of the second experiment, which is a summary of the evaluations of the six analyzed measures. Wu & Palmer (Sim'_{wup}) and Lin (Sim'_{lin}) metrics did not show any differences in performance for any β . The modifications to these metrics did not positively or negatively affect their performance. Both measures normalize the semantic commonalities and the semantic differences by the total amount of information gathered from the two concepts being compared (see Table 3). The correlations of the other modified metrics (Sim'_{length} , Sim'_{lch} , Sim'_{res} , $Sim'_{j\&c}$) were higher than those of the originals.

⁶ The Brown University Standard Corpus of Present-Day American English.

⁷ The original metric was not modified.

Table 2. Maximum values of correlation obtained for each measure

| | Original | Max Corr | β |
|--------|----------|----------|----------|
| length | 0.8401 | 0.8571 | 0.60 |
| wup | 0.7726 | 0.7726 | Original |
| lch | 0.8293 | 0.8296 | 12 |
| res | 0.8308 | 0.8676 | 2.8 |
| j&c | 0.8660 | 0.8672 | 2.8 |
| lin | 0.8587 | 0.8587 | Original |

Table 3. Normalization factor depending on the metric approach

| Metric | Approach | Normalization Factor |
|--------|------------|---------------------------|
| wup | edge-based | $depth(c_1) + depth(c_2)$ |
| lin | node-based | $IC(c_1) + IC(c_2)$ |

The new metrics obtained slightly better results when the semantic differences between the concepts were taken into consideration. In general, all node-based similarity measures were superior to the edge-based ones. Curiously, the results of Sim'_{length} were better than those for Sim'_{lch} despite simplicity of its model. The modified length metric (Sim'_{length}) reached its maximum correlation value at $\beta = 0.6$; increasing β further did not improve its correlation value. Leacock & Chodorow (Sim'_{lch}) reached its maximum correlation value at $\beta = 12$, although it was a small improvement over the values for the original function. Resnik (Sim'_{res}) and Jiang & Conrath ($Sim'_{j\&c}$) obtained their best correlation values for $\beta = 2.8$.

5 Concluding Remarks and Future Work

The five new measures presented in this paper are modifications of traditional WordNet-based semantic similarity metrics. Supported by a featured-based theory, they incorporate the idea of semantic differences between concepts into the similarity computation. The experimental results showed that, three of the measures outperformed their classical; the other two measures performed the same as their classical versions. These results demonstrate the strengths and positive effects of including concepts semantic differences during their semantic similarity computations.

As future work, we would like to find out why Sim'_{wup} and Sim'_{lin} metrics did not change their correlation coefficients after being modified. We will also evaluate the node-based metrics using the intrinsic information content approach. Additionally, we shall investigate the effect of the combination of commonalities and differences at the same time. And finally, we will work on a method for finding a best value for those parameters.

References

1. Budanitsky, A., Hirst, G.: Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In: Workshop on WordNet and Other Lexical Resources, 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (2001)

2. Fellbaum, C. (ed.): *Wordnet: An Electronic Lexical Database*, 1st edn. Bradford Books (1998)
3. Hai, J., Hanhua, C.: Semrex: Efficient search in a semantic overlay for literature retrieval. *Future Generation Computer Systems* 11(6), 475–488 (2008)
4. Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E.G.M., Milios, E.E.: Information retrieval by semantic similarity. *Int. Journal on Semantic Web and Information Systems (IJSWIS)* 2(3), 55–73 (2006)
5. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: *Int. Conf. on Research in Computational Linguistics*, pp. 19–33 (1997)
6. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. In: *WordNet: A Lexical Reference System and its Application*, pp. 265–283 (1998)
7. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering* 15(4), 871–882 (2003)
8. Lin, D.: An information-theoretic definition of similarity. In: *15th Int. Conf. on Machine Learning*, pp. 296–304 (1998)
9. Mazuel, L., Sabouret, N.: Semantic relatedness measure using object properties in an ontology. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) *ISWC 2008. LNCS*, vol. 5318, pp. 681–694. Springer, Heidelberg (2008)
10. Miller, G., Charles, W.: Contextual correlates of semantic synonymy. *Languages and Cognitive Processes* 6(1), 1–28 (1991)
11. Pirró, G., Seco, N.: Design, implementation and evaluation of a new semantic similarity metric combining features and intrinsic information content. In: Meersman, R., Tari, Z. (eds.) *OTM 2008, Part II. LNCS*, vol. 5332, pp. 1271–1288. Springer, Heidelberg (2008)
12. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics* 19(1), 17–30 (1989)
13. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: *Int. Joint Conf. on Artificial Intelligence*, vol. 14(1), pp. 448–453 (1995)
14. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Artificial Intelligence Research* 11, 95–130 (1999)
15. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. *Communications of ACM* 8(10), 627–633 (1965)
16. Seco, N.: *Computational models of similarity in lexical ontologies*. Master's thesis, University College Dublin (2005)
17. Tversky, A.: Features of similarity. *Psychological Review* 84(4), 327–352 (1977)
18. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: *32nd Annual Meeting of the Association for Computational Linguistics*, pp. 133–138 (1994)
19. Ziegler, C.-N., Simon, K., Lausen, G.: Automatic computation of semantic proximity using taxonomic knowledge. In: *15th ACM Int. Conf. on Information and Knowledge Management*, pp. 465–474 (2006)

An Ontological Representation of Documents and Queries for Information Retrieval Systems

Mauro Dragoni, Célia Da Costa Pereira, and Andrea G.B. Tettamanzi

Università degli Studi di Milano, Dipartimento di Tecnologie dell'Informazione
Via Bramante 65, 26013 Crema (CR), Italy
{mauro.dragoni,celia.pereira,andrea.tettamanzi}@unimi.it

Abstract. This paper presents a vector space model approach, for representing documents and queries, using concepts instead of terms and WordNet as a light ontology. This way, information overlap is reduced with respect to the classic semantic expansion techniques. Experiments carried out on the MuchMore benchmark showed the effectiveness of the approach.

1 Introduction

This paper presents an ontology-based approach to the conceptual representation of documents. Such an approach is inspired by a recently proposed idea presented in [9], and uses an adapted version of that method to standardize the representation of documents and queries. The proposed approach is somehow similar to query expansion [12]. However, additional considerations have been taken into account and some improvements have been applied as explained below.

Query expansion is an approach to boost the performance of Information Retrieval (IR) systems. It consists of expanding a query with the addition of terms that are semantically correlated with the original terms of the query. Several works demonstrated the improved performance of IR systems using query expansion [19,3,5]. However, query expansion has to be used carefully, because, as demonstrated in [8], expansion might degrade the performance of some individual queries. This is due to the fact that an incorrect choice of terms and concepts for the expansion task might harm the retrieval process by drifting it away from the optimal correct answer.

Document expansion applied to IR has been recently proposed in [2]. In that work, a sub-tree approach has been implemented to represent concepts in documents and queries. However, when using a tree structure, there is redundancy of information because more general concepts may be represented implicitly by using only the leaf concepts they subsume.

This paper presents a new representation for documents and queries. The proposed approach exploits the structure of the well-known WordNet machine-readable dictionary (MRD) to reduce the redundancy of information generally contained in a concept-based document representation. The second improvement

is the reduction of the computational time needed to compare documents and queries represented using concepts. This representation has been applied to the *ad-hoc* retrieval problem. The approach has been evaluated on the MuchMore¹ Collection [4] and the results demonstrate its viability.

The paper is organized as follows: in Section 2, an overview of the environment in which ontology has been used is presented. Section 3 presents the tools used for this work. Section 4 illustrates the proposed approach to represent information, while Section 5 compares this approach with other two well-known approaches used in conceptual representation of documents. In Section 6, the results obtained from the test campaign are discussed. Finally, Section 7 concludes.

2 Related Works

An increasing number of recent IR systems make use of ontologies to help the users clarify their information needs and come up with semantic representations of documents. Many ontology-based IR systems and models have been proposed in the last decade. An interesting review on IR techniques based on ontologies is presented in [11], while in [16] the author studies the application of ontologies to a large-scale IR system for web purposes. A model for the exploitation of ontology-based knowledge bases is presented in [7]. The aim of this model is to improve search over large document repositories. The model includes an ontology-based scheme for the annotation of documents, and a retrieval model based on an adaptation of the classic vector-space model [15]. Another IR system based on ontologies is presented in [14]. The authors propose an IR system which has a landmark information database with hierarchical structures and semantic meanings of the features and characteristics of the landmarks.

The implementation of ontology models has been also investigated by using fuzzy models [6].

In IR, queries entered by users usually are not detailed enough, making it hard to retrieve satisfactory results. Query expansion can help solve this problem. However, query expansion as usually implemented in IR systems does not guarantee consistent retrieval results. Ontologies play a key role in query expansion research. A common use of ontologies in query expansion is to enrich the resources with some well-defined meaning to enhance the search capabilities of existing web searching systems.

In [18], the authors propose and implement a query expansion method which combines a domain ontology with the frequency of terms. The ontology is used to describe domain knowledge; a logic reasoner and the frequency of terms are used to choose fitting expansion words. This way, higher recall and precision can be obtained.

In [10], the authors present an approach to expand queries whose idea is to look for terms from the topic query in an ontology to add similar terms.

¹ URL: <http://muchmore.dfki.de>

3 Preliminaries

The roadmap to prove the viability of a concept-based representation of documents and queries is composed of two main tasks:

- to choose a method that allows representing all document terms by using the same set of concepts;
- to implement an approach that allows indexing and evaluating each concept, in both documents and queries, with the “correct” weight.

To represent documents, the method described in Section 4 has been used, combined with the use of the WordNet MRD. From the WordNet database, the set of terms that do not have hyponyms has been extracted. We call such terms “base concepts”. A vector, named “base vector”, has been created and, to each component of the vector, a base concept has been assigned. This way, each term is represented by using the base vector of the WordNet ontology.

The representation described above has been implemented on top of the Apache Lucene open-source API.²

In the pre-indexing phase, each document has been converted into its ontological representation. After the calculation of the importance of each concept in a document, only concepts with a degree of importance higher than a fixed cut-off value have been maintained, while the others have been discarded. The cut-off value used in these experiments is 0.01. This choice has a drawback, namely that an approximation of representing information is introduced due to the discarding of some minor concepts. However, we have experimentally verified that this approximation does not affect the final results.

During the evaluation activity, queries have also been converted into the ontological representation. This way, weights have to be assigned to each concept to evaluate all concepts with the right proportion. One of the features of Lucene is the possibility of assigning a payload to each term of the query. Therefore, for each element in the concept-based representation of the query, the relevant concept weight has been used as boost value.

4 Document Representation

Conventional IR approaches represent documents as vectors of term weights. Such representations use a vector with one component for every significant term that occurs in the document. This has several limitations, including:

1. different vector positions may be allocated to the synonyms of the same term; this way, there is an information loss because the importance of a determinate *concept* is distributed among different vector components;
2. the size of a document vector must be at least equal to the total number of words of the language used to write the document;

² See URL <http://lucene.apache.org/>

3. every time a new set of terms is introduced (which is a high-probability event), all document vectors must be reconstructed; the size of a repository thus grows not only as a function of the number of documents that it contains, but also of the size of the representation vectors.

To overcome these weaknesses of term-based representations, an ontology-based representation has been recently proposed [9], which exploits the hierarchical *is-a* relation among concepts, i.e., the meanings of words. For example, to describe with a term-based representation documents containing the three words: “animal”, “dog”, and “cat” a vector of three elements is needed; with an ontology-based representation, since “animal” subsumes both “dog” and “cat”, it is possible to use a vector with only two elements, related to the “dog” and “cat” concepts, that can also implicitly contain the information given by the presence of the “animal” concept. Moreover, by defining an ontology base, which is a set of independent concepts that covers the whole ontology, an ontology-based representation allows the system to use fixed-size document vectors, consisting of one component per base concept.

Calculating term importance is a significant and fundamental aspect for representing documents in conventional IR approaches. It is usually determined through term frequency-inverse document frequency (TF-IDF). When using an ontology-based representation, such usual definition of term-frequency cannot be applied because one does not operate by keywords, but by concepts. This is the reason why it has been adopted the document representation based on concepts proposed in [9], which is a concept-based adaptation of TF-IDF.

The quantity of information given by the presence of concept z in a document depends on the depth of z in the ontology graph, on how many times it appears in the document, and how many times it occurs in the whole document repository. These two frequencies also depend on the number of concepts which subsume or are subsumed by z . Let us consider a concept x which is a descendant of another concept y which has q children including x . Concept y is a descendant of a concept z which has k children including y . Concept x is a leaf of the graph representing the used ontology. For instance, considering a document containing only “ xy ”, the occurrence of x in the document is $1 + (1/q)$. In the document “ xyz ”, the occurrence of x is $1 + (1/q(1 + 1/k))$. As it is possible to see, the number of occurrences of a leaf is proportional to the number of children which all of its ancestors have. Explicit and implicit concepts are taken into account by using the following formulas:

$$N(c) = \text{occ}(c) + \sum_{c \in \text{Path}(c, \dots, \top)} \sum_{i=2}^{\text{depth}(c)} \frac{\text{occ}(c_i)}{\prod_{j=2}^i |\text{children}(c_j)|}, \quad (1)$$

where $N(c)$ is the number of occurrences, both explicit and implicit, of concept c and $\text{occ}(c)$ is the number of lexicalizations of c occurring in the document.

Given the ontology base $I = b_1, \dots, b_n$, where the b_i s are the base concepts, the quantity of information, $\text{info}(b_i)$, pertaining to base concept b_i in a document is:

$$\text{info}(b_i) = \frac{N_{\text{doc}}(b_i)}{N_{\text{rep}}(b_i)}, \tag{2}$$

where $N_{\text{doc}}(b_i)$ is the number of explicit and implicit occurrences of b_i in the document, and $N_{\text{rep}}(b_i)$ is the total number of its explicit and implicit occurrences in the whole document repository. This way, every component of the representation vector gives a value of the importance relation between a document and the relevant base concept.

A concrete example can be explained starting from the light ontology represented in Figures 1 and 2, and by considering a document D_1 containing concepts “ $xyyzyz$ ”.

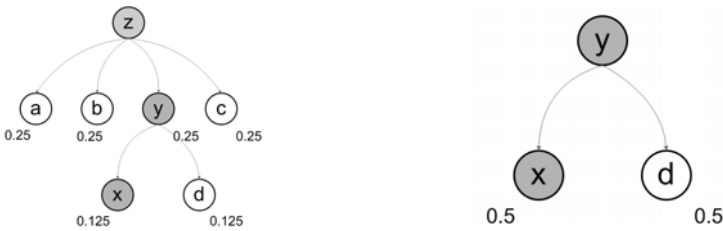


Fig. 1. Ontology representation for concept 'z'

Fig. 2. Ontology representation for concept 'y'

In this case, the ontology base is:

$$I = \{a, b, c, d, x\}$$

and, for each concept in the ontology, the information vectors are

$$\begin{aligned} \text{info}(z) &= (0.25, 0.25, 0.25, 0.125, 0.125), \\ \text{info}(a) &= (1.0, 0.0, 0.0, 0.0, 0.0), \\ \text{info}(b) &= (0.0, 1.0, 0.0, 0.0, 0.0), \\ \text{info}(c) &= (0.0, 0.0, 1.0, 0.0, 0.0), \\ \text{info}(y) &= (0.0, 0.0, 0.0, 0.5, 0.5), \\ \text{info}(d) &= (0.0, 0.0, 0.0, 1.0, 0.0), \\ \text{info}(x) &= (0.0, 0.0, 0.0, 0.0, 1.0), \end{aligned}$$

which yield the following document vector representation for D_1 :

$$D_1 = 2 \cdot \text{info}(x) + 3 \cdot \text{info}(y) + \text{info}(z) = (0.25, 0.25, 0.25, 1.625, 3.625). \tag{3}$$

In Section 5, a comparison between the proposed representation and other two classic concept-based representation is discussed.

5 Representation Comparison

In Section 4, the approach used to represent information was described. This section shows the improvements obtained by applying the proposed approach and illustrates a comparison between the proposed approach and other two approaches commonly used in conceptual document representation. The expansion technique is generally used to enrich the information content of queries. However, in the past years some authors applied the expansion technique also to represent documents [2]. Like in [13,2], we propose an approach that uses WordNet to extract concepts from terms.

The two main improvements obtained by the application of the ontology-based approach are illustrated below.

Information Redundancy. Approaches that apply the expansion of documents and queries use correlated concepts to expand the original terms of documents and queries. A problem with expansion is that information is redundant and there is no real improvement of the representation of the document (or query) content. With the proposed representation, this redundancy is eliminated, because only independent concepts are taken into account to represent documents and queries. Another positive aspect is that the size of the vector representing document content by using concepts is generally smaller than the size of the vector representing document content by using terms.

An example of a technique that shows this drawback is presented in [13]. In this work the authors propose an indexing technique that takes into account WordNet synsets instead of terms. For each term in documents, the synsets associated to that terms are extracted and then used as token for the indexing task. This way, the computational time needed to perform a query is not increased, however, there is a significant overlap of information because different synsets might be semantically correlated. An example is given by the terms “animal” and “pet”: these terms have two different synsets; however, observing the WordNet lattice, the term “pet” is linked with an *is-a* relation to the term “animal”. Therefore, in a scenario in which a document contains both terms, the same conceptual information is repeated. This is clear, because, even if the terms “animal” and “pet” are not represented by using the same synset, they are semantically correlated, since “pet” is a sub-concept of “animal”. This way, when a document contains both terms, the presence of the term “animal” has to contribute to the importance of the concept “pet” instead of being represented with a different token.

Computational Time. When IR approaches are applied in a real-world environment, the computational time needed to evaluate the match between documents and the submitted query has to be considered. It is known that systems using the vector space model have higher efficiency. Conceptual-based approaches, such as the one presented in [2], generally implement a non-vectorial data structure which needs a higher computational time with respect to a vector space model representation. The approach proposed in this paper overcomes this issue

because the document content is represented by using a vector and, therefore, the computational time needed to compute document scores is comparable to the computational time needed when using the vector space model.

6 Experiments

In this section, the impact of the ontology document and query representation is evaluated. The evaluation method follows the TREC protocol [17]. For each query, the first 1000 retrieved documents have been considered and the precision of the system has been calculated at different points: 5, 10, 15, and 30 documents retrieved. Moreover, the precision/recall graph has been calculated.

The experimental campaign has been performed by using the MuchMore collection, that consists of 7,823 abstracts of medical papers and 25 queries with their relevance judgments. One of the particular features of this collection is that there are numerous medical terms. This gives an advantage to term-based representations over the semantic representation, because specific terms present in documents (e.g., “arthroscopic”) are very discriminant. Indeed, by using a semantic expansion, some problems may occur because, generally, the MRD and thesaurus used to expand terms do not contain all of the domain-specific terms.

The precision/recall graph shown in Figure 3 illustrates the comparison between the proposed approach (gray curve with circle marks), the classical term-based representation (black curve), and the synset representation method [13] (light gray curve with square marks). As expected, for all recall values, the proposed approach obtained better results than the term-based and synset-based

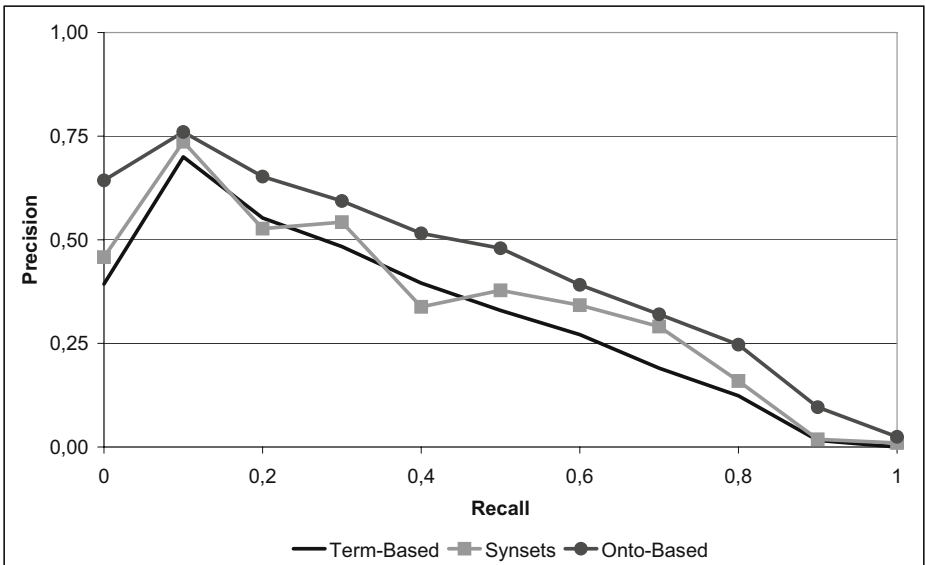


Fig. 3. Precision/recall results

representations. The best gain over the subset-based representation is at recall levels 0.0, 0.2, and 0.4, while, for recall values between 0.6 and 1.0, the synset-based precision curve lies within the other two curves.

A possible explanation for this scenario is that, for documents that are well related to a particular topic, the adopted ontological representation is able to improve the representation of the documents contents. However, for documents that are partially related to a topic or that contain many ambiguous terms, the proposed approach becomes less capable of maintaining a high precision. At the end of this section, some improvements that may help overcome this issue are discussed.

In Table 1, the three different representations are compared with respect to the Precision@X and MAP values. The results show that the proposed approach obtains better results for all the precision levels and also for the MAP value.

Table 1. Comparisons table between semantic expansion approaches

| Systems | Precisions | | | | |
|----------------------|------------|-------|-------|-------|-------|
| | P5 | P10 | P15 | P30 | MAP |
| Term-Based | 0.544 | 0.480 | 0.405 | 0.273 | 0.449 |
| Synset-Indexing [13] | 0.648 | 0.484 | 0.403 | 0.309 | 0.459 |
| Concept-Based | 0.744 | 0.544 | 0.478 | 0.394 | 0.507 |

An in-depth study of this first experiments campaign has been performed, and we have noticed that for some queries the concept-based representation obtained results that were below our expectations. By inspecting the implemented model, some issues have been noticed and are at present under analysis:

- Absence of some terms in the ontology: some terms, in particular terms related to specific domains (biomedical, mechanical, business, etc.), are not defined in the MRD used to define the concept-based version of the documents. This way there is, in some cases, a loss of information that affects the final retrieval result.
- Proper names have not been considered: proper names of persons, geographical locations, industries, etc., are not present in the concept-based index. Observing the content of some documents and topics, proper names turn out to be a discriminant feature in some cases. However, this may be interpreted as an “instance” issue: when a domain is specified, our method might incorporate domain-related instances that suggest relevant concepts.
- Verbs and adjective are not present as well in the ontology: the concept representation of terms, described in Section 4, does not take into account verbs and adjectives. This happens because verbs and adjectives are structured in a different way than nouns. The hyperonymy and hyponymy relations (that make MRD comparable with ontologies) are not defined for verbs and adjectives. However, a method mapping verbs and adjective to their related nouns is being implemented to overcome this drawback.

- Term ambiguity: the concept-based representation has the problem of introducing an error given by not using a word-sense disambiguation (WSD) algorithm. Using such a method, concepts associated to incorrect senses would be discarded or weighted less. Therefore, the concept-based representation of each word would be finer, with the consequence of representing the information contained in a document with higher precision.

Improving the actual model with the above suggestions would certainly yield significantly better results in the future experimental campaign. This positive view is motivated by the fact that, in spite of these issues, the preliminary goal of outperforming the precision of the term-based representation has been accomplished.

7 Conclusion

In this paper, we have discussed an approach to indexing documents and representing queries for IR purposes which exploits a conceptual representation based on ontologies.

Experiments have been carried out on the MuchMore Collection to validate the approach with respect to problems like term-synonymity in documents.

Preliminary experimental results show that the proposed representation improves the ranking of the documents. Investigation on results highlights that further improvement could be obtained by integrating WSD techniques like the one discussed in [1] to avoid the error introduced by considering incorrect word senses, and with a better usage and interpretation of WordNet to overcome the loss of information caused by the absence of proper nouns, verbs, and adjectives.

References

1. Azzini, A., Dragoni, M., da Costa Pereira, C., Tettamanzi, A.: Evolving neural networks for word sense disambiguation. In: Proc. of HIS 2008, Barcelona, Spain, September 10-12, pp. 332–337 (2008)
2. Baziz, M., Boughanem, M., Pasi, G., Prade, H.: An information retrieval driven by ontology: from query to document expansion. In: Evans, D., Furui, S., Soulé-Dupuy, C. (eds.) RIAO. CID (2007)
3. Billerbeck, B., Zobel, J.: Techniques for efficient query expansion. In: Apostolico, A., Melucci, M. (eds.) SPIRE 2004. LNCS, vol. 3246, pp. 30–42. Springer, Heidelberg (2004)
4. Boughanem, M., Dkaki, T., Mothe, J., Soulé-Dupuy, C.: Mercure at trec7. In: Lamersdorf, W., Merz, M. (eds.) TREC 1998. LNCS, vol. 1402, pp. 355–360. Springer, Heidelberg (1998)
5. Cai, D., van Rijsbergen, C., Jose, J.: Automatic query expansion based on divergence. In: CIKM, pp. 419–426. ACM, New York (2001)
6. Calegari, S., Sanchez, E.: A fuzzy ontology-approach to improve semantic information retrieval. In: Bobillo, F., da Costa, P., d'Amato, C., Fanizzi, N., Fung, F., Lukasiewicz, T., Martin, T., Nickles, M., Peng, Y., Pool, M., Smrz, P., Vojtás, P. (eds.) URSW. CEUR Workshop Proceedings, vol. 327 (2007), CEUR-WS.org

7. Castells, P., Fernández, M., Vallet, D.: An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Trans. Knowl. Data Eng.* 19(2), 261–272 (2007)
8. Cronen-Townsend, S., Zhou, Y., Croft, W.: A framework for selective query expansion. In: Grossman, D., Gravano, L., Zhai, C., Herzog, O., Evans, D. (eds.) *CIKM*, pp. 236–237. ACM, New York (2004)
9. da Costa Pereira, C., Tettamanzi, A.G.B.: An ontology-based method for user model acquisition. In: Ma, Z. (ed.) *Soft computing in ontologies and semantic Web. Studies in fuzziness and soft computing*, pp. 211–227. Springer, Heidelberg (2006)
10. Díaz-Galiano, M., Cumbreiras, M.G., Martín-Valdivia, M., Ráez, A.M., Ureña-López, L.: Integrating mesh ontology to improve medical information retrieval. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) *CLEF 2007. LNCS*, vol. 5152, pp. 601–606. Springer, Heidelberg (2008)
11. Dridi, O.: Ontology-based information retrieval: Overview and new proposition. In: Pastor, O., Flory, A., Cavarero, J.-L. (eds.) *RCIS*, pp. 421–426. IEEE, Los Alamitos (2008)
12. Efthimiadis, E.: Query expansion. In: Williams, M.E. (ed.) *Annual review of information science and technology*, vol. 31, pp. 121–187. Information Today Inc., Medford (1996)
13. Gonzalo, J., Verdejo, F., Chugur, I., Cigarrán, J.M.: Indexing with wordnet synsets can improve text retrieval. *CoRR cmp-lg/9808002* (1998)
14. Hattori, T., Hiramatsu, K., Okadome, T., Parsia, B., Sirin, E.: Ichigen-san: An ontology-based information retrieval system. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) *APWeb 2006. LNCS*, vol. 3841, pp. 1197–1200. Springer, Heidelberg (2006)
15. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. *Commun. ACM* 18(11), 613–620 (1975)
16. Tomassen, S.: Research on ontology-driven information retrieval. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops. LNCS*, vol. 4278, pp. 1460–1468. Springer, Heidelberg (2006)
17. Voorhees, E., Harman, D.: Overview of the sixth text retrieval conference (trec-6). In: *TREC*, pp. 1–24 (1997)
18. Wu, F., Wu, G., Fu, X.: Design and implementation of ontology-based query expansion for information retrieval. In: Xu, L., Tjoa, A., Chaudhry, S. (eds.) *CONFENIS (1). IFIP*, vol. 254, pp. 293–298. Springer, Heidelberg (2007)
19. Xu, J., Croft, W.: Query expansion using local and global document analysis. In: Frei, H.-P., Harman, D., Schäuble, P., Wilkinson, R. (eds.) *SIGIR*, pp. 4–11. ACM, New York (1996)

Predicting the Development of Juvenile Delinquency by Simulation

Tibor Bosse, Charlotte Gerritsen, and Michel C.A. Klein

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{tbosse,cg,mcaklein}@few.vu.nl

Abstract. A large number of delinquent activities are performed by adolescents and only occur during this period in their lives. One of the main factors that influence this behaviour is social interaction, mainly with peers. This paper contributes a computational model that predicts delinquent behaviour during adolescence based on interaction with friends and classmates. Based on the model, which was validated based on empirical data, the level of delinquency of pupils is simulated over time. Furthermore, simulation experiments are performed to investigate for hypothetical scenarios what is the impact of the division of students over classes on the (individual and collective) level of delinquency.

Keywords: social simulation, social learning, delinquent behaviour.

1 Introduction

One of the main challenges in Criminology is to understand, explain and predict when individuals show delinquent behaviour [4]. Obviously, there is a wide range of potential contributors to the emergence of crime, varying from environmental opportunities to social influences. In this paper we focus on the latter. Learning (delinquent) behaviour by social interaction is something that is often observed in adolescents [11]. During the period from 12 to 18 year old, people are more susceptible to the opinion of their peers. In some situations, their desire to be part of a group can be so strong that they break some rules to achieve this desire. This is consistent with the theory by Moffitt [10] who states that one can divide delinquents roughly into two groups, namely life-course persistent offenders and adolescence limited offenders. The behaviour of the first group is caused by neuropsychological problems during childhood that interact cumulatively with their criminogenic environments across development, which leads to a pathological personality. This behaviour will usually continue through life. Instead, the behaviour of adolescence-limited offenders is caused by a gap between biological maturity and social maturity. It is mainly caused by mimicking antisocial role models like peers, but also parents and school are important contributors. These offenders peak sharply at about age 17 and drop fast in young adulthood.

In this paper we exploit simulation techniques to study the development of such juvenile delinquency. As mentioned above, this type of behaviour is limited to a certain period of time, and some of its direct causes are clearly determined. This provides

opportunities to develop a computational model of this process. In previous research [2], we developed such a model, which was able to predict the level of delinquency of students based on information about the personal characteristics and their peer network. The model was validated by using a large dataset with information about 1730 scholars (taken from [14]).

The main contribution of the current paper is to show how this model can be used to perform so called *what-if simulation experiments*. In these simulations the existing (validated) model is applied to a hypothetical situation, which is slightly different from the situation in the existing empirical data. For example, we want to see what happens to the level of delinquency (both of individuals and of the classes) when the composition of the classes is altered. Interesting questions here are, among others:

- What is the effect when we put the most delinquent students together in one class?
- Is it better to spread the delinquent and non-delinquent students equally over classes?

To answer such questions, this paper proposes to make use of social simulation techniques [3]. In recent years, a number of papers have successfully tackled criminological questions using social simulation, e.g., [7, 9]. However, the current paper differs from these approaches in that we do not attempt to reproduce existing data, but rather explore how hypothetical scenarios would evolve. We will create these hypothetical scenarios by making small modifications in existing scenarios (e.g., change the composition of classes), and run the simulation model on the modified data. The main question that we would like to answer is whether the composition of a school class has an influence on the overall level of delinquency of the pupils. This is an interesting topic, since it is often believed that the structure of schools and peer networks has an important impact on juvenile delinquency [8, 12].

The paper is organised as follows. In Section 2 we describe how the data used for the simulation experiments were collected. The simulation model itself is presented in Section 3, and the experiments in Section 4. Finally, Section 5 concludes the paper with a discussion and some ideas for future work.

2 Data Collection

The model presented in this paper is based on empirical data from a longitudinal research project. This research was performed by the Netherlands Institute for the Study of Crime and Law Enforcement (NSCR) in the so called ‘School Project’ [14], which focused on peer network formation, personal development, and school interventions in the development of problem behaviour and delinquency.

In this project, a large number of high school students were surveyed by means of questionnaires. As respondents, a cohort of students was used that started high school during the school year 2001/2002. The first year of secondary education in the Netherlands is comparable with 7th grade in the United States (most students are 12 or 13 years old). These students were surveyed during three consecutive years: 2002, 2003 and 2004.

During these three years, the respondents had to fill out a number of questionnaires. Their delinquent behaviour was measured using self-reports of a variety of offences. The self report method is a standard procedure in Criminology, and it results

in fairly reliable estimates of delinquency levels of young people, when it is conducted in a proper way and in an anonymous setting. Respondents were asked if they had ever committed an offence and, if so, how often during the reference period. The measures of self-reported delinquency used in this study come from 12 questions, among which: in the last year, how many times did you: “paint graffiti”, “vandalise property”, or “steal small things from shops worth less than 5 Euros” The total delinquency measure indicates how many types of the 12 possible types of delinquent behaviours were reported by the person.

The respondents also had to answer a number of questions about their friends (e.g. with whom they spent a lot of time, who were their best friends), to obtain information about their social networks. In the analyses, friends’ numbers were linked to the respondent’s own number, enabling the networks of friends to be mapped and analysed.

Further, the study also used a substantial number of other measures on risk factors that are central in criminological theories and have been found to correlate with delinquency in the past (e.g. low supervision and support by parents, low bond with school, low law conformity, high impulsivity, high temper). For more details of the empirical research see [13, 14].

3 Simulation Model

In this section the simulation model used for the experiments is described. First, in Section 3.1, the methodology behind the design of the model is discussed (based on [1]). Section 3.2 presents the implementation of the model, and Section 3.3 shows how the original model was extended in order to incorporate information about classes.

3.1 Design Methodology

As a first step in the process of designing the model, an initial dynamic model was developed for the development of delinquency through social learning in a classroom, based on an analysis of the literature (see Figure 1). A more detailed description of this model is provided in [1]. The model describes the influences of several personal characteristics, as well as the influences of other peers. More specifically, the delinquency of an agent is influenced by its previous delinquency, its individual personality traits (e.g. temper, impulsiveness), and external factors (i.e., the school, the parents, and peers). This original model has the form of a set of differential equations, where delinquency is measured as a real number between 0 and 1. In [1], it has been shown that this model can be used to simulate delinquency development of a small set of agents in a classroom. The simulations exhibited several patterns that would be expected based on the criminological literature.

A next step was to validate the model based on the empirical data mentioned in Section 2. In this research [2], a representative sample of the collected dataset has been selected, and has been split up in a training set and test set. Each set contained the data of around 250 pupils. A lot of pupils were left out of the original dataset, because their questionnaires were not suitable. This was caused, for instance, by gaps in the answers or because they were only attending a particular school during part of the research period. When making this split, we guaranteed that there was no overlap between the schools

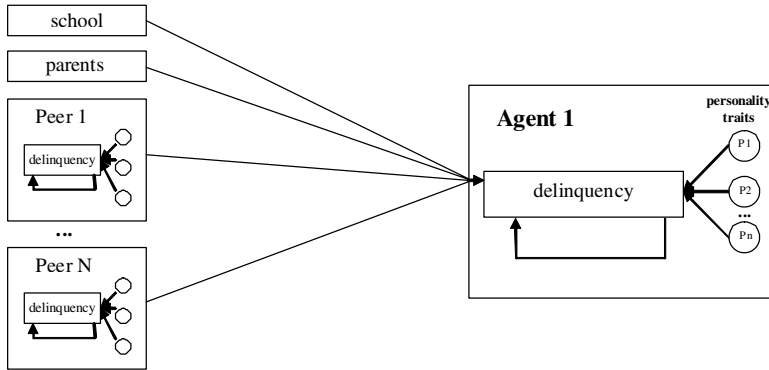


Fig. 1. Overview of the simulation architecture (from [1])

used in the training set and those used in the test set. We developed an evaluation method that could be used to quantify the correctness of models and to discriminate between accurate and less accurate models. This measure accommodates the intuitive ideas about a correct prediction in one number (see Section 3.2). The model was calibrated with the data in the training set by taking the model from [1] extended with some additional factors reported in [14], and systematically adjusting it and comparing the simulation results with the actual measurements in the training set (scaled to a number between 0 and 1). The adjustment consisted of both ignoring factors in the model (i.e. leaving out variables in the formulae) and calibrating parameters (i.e. changing the value of weighting variables), thereby creating different variations of the model.

Finally, the second data set was used to validate the different variations of the model that seemed promising during the calibration phase. In this phase, we did not change the model or parameters, but just calculated the accuracy according the developed measure for all formulae that resulted in a high score in the first phase. This method gives an unbiased validation of the accuracy, as the validation is performed on a different data set than the tuning.

3.2 Implementation

To implement the model, we used standard numerical simulation software. A ‘school class’ was modelled as a multi-dimensional array, where each array represented a different student. The different dimensions represented characteristics of the students over time. For example, these dimensions specified the individual characteristics (like impulsivity and risk-orientedness) and the relations to peers. To calculate the new delinquency of each agent, the following algorithm was used (in pseudo code):

For each agent:

1. determine current delinquency
2. determine individual characteristics
3. compose the social network (friends)
4. calculate average delinquency of social network
5. calculate new delinquency, using information from step 1, 2, and 4

To calculate the new delinquency of the individual agents (step 5), various variants of the model have been tried, each incorporating some of the factors identified in the previous section. These different models are depicted in Table 1. For example, model variant 1 (a baseline model), always predicts that students will not become delinquent. The last column denotes the accuracy rate for each model, which was calculated as follows:

$$\text{Accuracy Rate} = \frac{(w \cdot \text{Hits} + \text{Correct Rejections})}{(w \cdot \text{Hits} + w \cdot \text{Misses} + \text{False Alarms} + \text{Correct Rejections})}$$

where Hits, Misses, Correct Rejections and False Alarms are defined according to the classical measures in signal detection theory [5]. For more details, see [2]. The factor ‘risk orientedness’ (model 8 and 9) indicates the extent to which the pupils like performing exciting activities, and the factor ‘deviance reinforcement’ (model 9-11) indicates the extent to which the pupils are sensitive to influences of their friends.

As can be seen, variants 10 and 11 have the highest accuracy. This means that the previous delinquency combined with the impulsivity, the level of deviance reinforcement by friends, and the delinquency of (best) friends, seem to be the best predictors for delinquent behaviour.

Table 1. Variants of the model and accuracy values (taken from [2])

| Model variant | Main factors used | Accuracy |
|---------------|-----------------------------------------------------------------------------------------------------------|----------|
| 1 | always predict non-delinquency | 45.42 |
| 2 | always predict delinquency | 54.58 |
| 3 | randomly predict non-delinquency and delinquency (with ratio 1:1) | 50.31 |
| 4 | randomly predict non-delinquency and delinquency (with ratio 3:1) | 48.16 |
| 5 | all factors identified in Section 3.1 | 66.10 |
| 6 | delinquency last year | 70.52 |
| 7 | delinquency last year, delinquency friends | 71.04 |
| 8 | delinquency last year, delinquency friends, risk-orientedness, temper | 72.64 |
| 9 | delinquency last year, risk-orientedness, deviance reinforcement, delinquency friends | 73.62 |
| 10 | delinquency last year, impulsivity, deviance reinforcement, delinquency best friends | 76.41 |
| 11 | delinquency last year, impulsivity, deviance reinforcement, delinquency friends, delinquency best friends | 76.24 |

In addition to the accuracy, the quality of the models has also been tested using a Relative Operating Characteristics (ROC) analysis. This method has been used because this is a standard measure in the literature and allows us to compare the results with studies in other domains. The outcome of this analysis is a curve which represents a graphical plot of the fraction of true versus the fraction of false positives for a binary classifier system as its *discrimination threshold* is varied (see Figure 1). The threshold in our model is the value of the calculated delinquency above which a pupil is classified as delinquent. We calculated the area under the ROC curve (AUC), a scalar measure for the quality of the predictions, for each model. For model variant

10, the AUC is 0.79. An AUC-value larger than 0.70 is called ‘acceptable’, larger than 0.80 ‘excellent’ and larger than 0.90 ‘outstanding’ [6]¹.

3.3 Incorporating Class Information

Although model 10 and 11 produce the highest accuracy rates, these model variants are not particularly appropriate for the aims of the current paper. That is, the goal of this paper is to predict for hypothetical scenarios (which are slightly different from the existing situation) how the delinquency of the students would have developed. And since it is not very realistic to assume that one can easily modify, say, the impulsivity or the friend network of students, variant 10 and 11 are not very useful candidates for these ‘what-if experiments’.

For this reason, two additional variants of the model have been developed. These models (variant 12 and 13) use the composition of classes. For obvious reasons, in practice it is much easier to manipulate students’ class composition than their friend networks. Therefore this factor was also manipulated within the hypothetical scenarios. To this end, variants of the model have been developed that take the delinquency of class members into account.

Model variant 12 predicts that a student will become delinquent if (s)he was delinquent in the previous year OR (s)he is part of a delinquent class AND (s)he has a high value for ‘deviance reinforcement’. Here, being part of a delinquent class is defined as the situation that the average delinquency of all students in the class is higher than a certain threshold. Note that this variant does not make use of the friend network.

The ROC curve obtained for this model variant 12 is depicted in Figure 2a, when compared with a random model (variant 3). As can be seen, variant 12 performs much better than the random model. The AUC of model variant 12 was 0.734, and its accuracy is 72.33. Although this is lower than the AUC and accuracy of variant 10 (resp. 0.79 and 76.41), we decided to use variant 12 for the simulation experiments described in the next section, because (as explained above) this variant contains the students’ classes as one of the factors.

In addition, a model variant has been developed that also takes the delinquency of the friends into account. Variant 13 predicts that a student will become delinquent if (s)he was delinquent in the previous year OR delinquency of the friends times the ‘deviance reinforcement’ is higher than a certain threshold OR (s)he is part of a delinquent class AND (s)he has a high value for ‘deviance reinforcement’. For being part of a delinquent class the same definition is used as in variant 12. The AUC of this model variant (see Figure 2b) is 67.52², and its accuracy is 72.66.

¹ Note that the AUC approach both has advantages and drawbacks when compared to the accuracy approach. An advantage is that this measure is rather common in the literature, which makes it easier to interpret the numbers, and to compare them with other models. A drawback is that the resulting numbers are calculated on the basis of all possible instances of the discrimination threshold, whereas in the accuracy approach only the best instance is taken. And since for the simulation experiments only this best instance will be used, the accuracy approach could be considered to be more useful.

² This relatively low number is mainly due to the dip at the right-hand side of the graph. This dip is caused by the fact that for some extreme values (which will obviously not be used in the simulation experiments) of the discrimination threshold, the model scores very bad. For this reason, in this case the accuracy may be more informative (see also footnote 1).

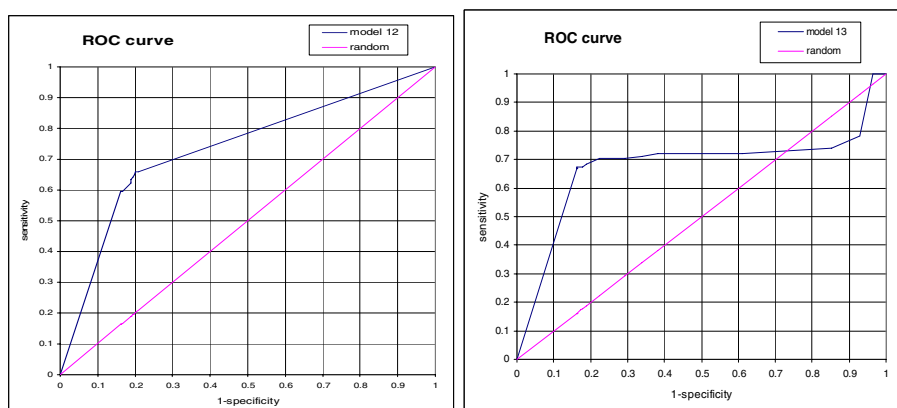


Fig. 2. ROC curves for a) model 12 and b) model 13 against a random prediction

4 Simulation Experiments

This section describes the simulation experiments that were performed to investigate the development of the delinquency of the pupils in the dataset for hypothetical scenarios. In Section 4.1, the setup of the experiments is explained. The results of the experiments are discussed in Section 4.2.

4.1 Approach

In the simulations, we compared the results of the simulation of the delinquency over one year using the actual class composition with the results of two simulations using a hypothetical composition, namely 1) a scenario in which all delinquent pupils are put together in the same class, and 2) a scenario in which all delinquent pupils are evenly distributed over all classes in a school. The goal of the comparison is to find out whether the change in the delinquency of pupils is positively or negatively influenced by the class composition.

The simulations are performed using three schools in our dataset, consisting of 6, 8 and 4 classes, respectively. These schools were chosen because many pupils of these schools filled out the questionnaire, so much data was available. In total 194 pupils were involved in the simulations. The simulations for the actual class composition and the two hypothetical scenarios have been performed two times, using each variant of the model that takes the class information into account (variant 12 and 13).

4.2 Simulation Results

Table 2 gives an overview of the development of the delinquency over a year according to the simulation with model variant 12 and 13. The first two columns indicate, respectively, the code of the school class in the study (e.g., '1 - 2' stands for 'class 2 of school 1'), and the amount of pupils in the class. In the next 3 columns, the 'base

before' column shows the number of delinquent pupils in the actual class composition at the start, and the columns 'base after v12' / 'v13' the predicted number of delinquent pupils after a year using model variant 12 or 13 respectively. Similarly, the 6 subsequent columns show the number of delinquent pupils in a class at the start and the end using the hypothetical class compositions (called scenario 1 and 2). It can be seen that in scenario 1 all delinquent pupils of a school are put together in a class, while in scenario 2 the delinquent pupils are more or less evenly distributed over the classes.

Table 2. Results of the simulations of delinquency of pupils with alternative class compositions using model variant 12 and 13

| school class | class size | base before | base after v12 | base after v13 | scen1 before | scen1 after v12 | scen1 after v13 | scen2 before | scen2 after v12 | scen2 after v13 |
|--------------|------------|-------------|----------------|----------------|--------------|-----------------|-----------------|--------------|-----------------|-----------------|
| 1 - 1 | 7 | 0 | 0 | 0 | 6 | 6 | 6 | 1 | 1 | 1 |
| 1 - 2 | 17 | 2 | 2 | 2 | 0 | 0 | 0 | 1 | 1 | 2 |
| 1 - 3 | 10 | 2 | 2 | 2 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 - 4 | 6 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 - 5 | 6 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 - 6 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 2 - 1 | 9 | 2 | 3 | 3 | 9 | 9 | 9 | 5 | 6 | 6 |
| 2 - 2 | 17 | 7 | 9 | 9 | 17 | 17 | 17 | 5 | 7 | 7 |
| 2 - 3 | 11 | 6 | 6 | 8 | 9 | 10 | 10 | 5 | 5 | 7 |
| 2 - 4 | 11 | 6 | 6 | 7 | 0 | 0 | 1 | 5 | 5 | 6 |
| 2 - 5 | 10 | 3 | 3 | 5 | 0 | 0 | 3 | 5 | 5 | 7 |
| 2 - 6 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 - 7 | 17 | 9 | 9 | 10 | 0 | 0 | 4 | 5 | 5 | 6 |
| 2 - 8 | 13 | 2 | 2 | 2 | 0 | 0 | 1 | 5 | 6 | 6 |
| 3 - 1 | 6 | 1 | 1 | 1 | 6 | 6 | 6 | 3 | 3 | 3 |
| 3 - 2 | 10 | 2 | 2 | 2 | 7 | 7 | 7 | 3 | 3 | 3 |
| 3 - 3 | 21 | 7 | 7 | 7 | 0 | 0 | 0 | 4 | 4 | 4 |
| 3 - 4 | 16 | 3 | 4 | 4 | 0 | 0 | 1 | 3 | 3 | 4 |
| total | 194 | 54 | 58 | 64 | 54 | 55 | 67 | 54 | 58 | 66 |

As can be seen in Table 2, the difference between the baseline and the different scenarios is not very high. For model variant 12, the total number of delinquent pupils increases in scenario 1 from 54 to 55 instead of to 58 for the baseline, and in scenario 2 it increases as much as in the baseline. In model variant 13, the number of delinquent pupils increases to 64 in the baseline, while it increases to 67 in scenario 1 and to 66 in scenario 2.

In the simulations using variant 12 we see that the increase of the number of delinquent pupils is less for the scenario in which all bad guys are put together (scenario 1) than in the baseline scenario or the scenario in which the delinquent pupils are evenly distributed. However, this pattern is not visible when using model variant 13. Overall, the differences between the results of the baseline scenario and the two other scenarios are very small. Although care should be taken not to draw too strict conclusions from these preliminary experiments, this may be an indication that the use of alternative class compositions has little effect.

5 Conclusion

In this paper, we have presented a number of simulation experiments on juvenile delinquency. The simulations were performed using an existing model that was based on the theory of social learning. In our previous research we have used empirical data about juvenile delinquency and social networks to develop and validate this simulation model. In the current paper we have presented some novel variants of this model. Moreover, we have used the model to investigate the effect of different class compositions on the development of the delinquency in the total group of pupils.

The experiments show no significant difference between the change in the total number of delinquent pupils in the different scenarios. The two different scenarios represented two extreme situations: all delinquent pupils put together, or all delinquent pupils distributed over all classes. Therefore, our tentative conclusion is that the composition of classes has not so much effect on the overall development of the delinquency of the pupils in a school. This is an interesting finding, since it is often argued that careful composition of school classes is very important to prevent development of juvenile delinquency [8, 12].

However, there are a few remarks that can be made about our experiments, which could be of influence on this conclusion. First of all, the model is possibly not very precise (see the relatively limited accuracy) because of small size of the training set. It could be the case that with a more precise model (derived from a larger training set) stronger effects would be visible. A second remark concerns the size of the classes. The ones used in the simulated scenarios are much smaller than regular classes; as a consequence, the influence of other pupils in the class is smaller in our simulations than in reality. The class size is this small because the data of many of the pupils was not suitable, e.g., because of missing information or because they switched between schools. The fact that we do not see a clear effect could also be caused by the fact that the number of offenders in our data set is relatively small. Therefore, also the number of predicted changes will always be quite small. Finally, we want to remark that the current models do not allow pupils to learn non-delinquency from their peers at school, they can become delinquent. Although this apparently follows from our dataset in the best predictive models, it might be the case that this is a bit different in reality.

Despite these remarks, the approach presented in this paper has proved to be a useful additional tool for criminology scientists, as also confirmed by our colleagues in the Criminology department. The approach allows for experiments that can not be easily performed in the real world and could give some indication of the expected effects of class compositions on juvenile delinquency.

Acknowledgement

The authors are very grateful to Frank Weerman of the Netherlands Institute for the Study of Crime and Law Enforcement for his willingness to provide the empirical data from the ‘School Project’, and for a number of fruitful discussions.

References

1. Bosse, T., Gerritsen, C., Klein, M.C.A.: Agent-Based Simulation of Social Learning in Criminology. In: Proc. of the Int. Conf. on Agents and AI, ICAART 2009, pp. 5–13. INSTICC Press (2009)
2. Bosse, T., Gerritsen, C., Klein, M.C.A., Weerman, F.M.: Development and Validation of an Agent-Based Simulation Model of Juvenile Delinquency. In: Proc. of the Int. Symposium on Social Intelligence and Networking, SIN 2009, pp. 200–207. IEEE Computer Society Press, Los Alamitos (2009)
3. Davidsson, P.: Agent Based Social Simulation: A Computer Science View. *Journal of Artificial Societies and Social Simulation* 5(1) (2002)
4. Gottfredson, M., Hirschi, T.: *A General Theory of Crime*. Stanford University Press (1990)
5. Green, D.M., Swets, J.A.: *Signal Detection Theory and Psychophysics*. Wiley, NY (1966)
6. Hosmer, D., Lemeshow, S.: *Applied logistic Regression*. John Wiley & Sons, NY (2000)
7. Liu, L., Wang, X., Eck, J., Liang, J.: Simulating Crime Events and Crime Patterns in RA/CA Model. In: Wang, F. (ed.) *Geographic Information Systems and Crime Analysis*, pp. 197–213. Idea Group, Singapore (2005)
8. Matsueda, R.L., Anderson, K.: The dynamics of delinquent peers and delinquent behaviour. *Criminology* 36, 269–308 (1998)
9. Melo, A., Belchior, M., Furtado, V.: Analyzing Police Patrol Routes by Simulating the Physical Reorganisation of Agents. In: Sichman, J.S., Antunes, L. (eds.) *MABS 2005. LNCS (LNAI)*, vol. 3891, pp. 99–114. Springer, Heidelberg (2006)
10. Moffitt, T.E.: Adolescence-Limited and Life-Course-Persistent Antisocial Behavior: A Developmental Taxonomy. *Psych. Review* 100(4), 674–701 (1993)
11. Sutherland, E.H., Cressey, D.R.: *Principles of Criminology*, 7th edn. J.B. Lippincott, Philadelphia (1966)
12. Warr, M.: *Companions in Crime. The social aspects of criminal conduct*. Cambridge University Press, Cambridge (2002)
13. Weerman, F.M., Bijleveld, C.C.J.H.: Birds of Different Feathers. *European Journal of Criminology* 4(4), 357–383 (2007)
14. Weerman, F.M., Smeenk, W., Harland, P. (eds.): *Problem behavior of students during secondary education: Individual development, student networks and reactions from school (in Dutch)*. Aksant, Amsterdam (2007)

Building and Analyzing Corpus to Investigate Appropriateness of Argumentative Discourse Structure for Facilitating Consensus

Tatiana Zidrasco^{1,2}, Shun Shiramatsu¹, Jun Takasaki¹, Tadachika Ozono¹,
and Toramatsu Shintani¹

¹ Department of Computer Science and Engineering, Nagoya Institute of Technology,
Gikiso-cho, Showa-ku, Nagoya, Aichi, Japan

² Applied Informatics Department, Technical University of Moldova,
Stefan cel Mare av. 168, 2068 Chisinau, Moldova

{tatiana,siramatu,takajun,ozono,tora}@toralab.ics.nitech.ac.jp

Abstract. Clarifying characteristics of *appropriate* argumentative discourse is important for developing computer assisted argumentation systems. We describe the analysis of argumentative discourse structure on the basis of Rhetorical Structure Theory in order to clarify what kind of argumentative discourse structure should be considered *appropriate*. We think that there exist specific agreement-oriented sequences of rhetorical relations in argumentative discourse that tend to lead to an agreement. We build a small argumentative corpus annotated with rhetorical relations and calculate posteriori probability for rhetorical relations bigrams to investigate what rhetorical relations precede *agreement*.

Keywords: corpus analysis, argumentative discourse, consensus building, rhetorical structure theory.

1 Introduction

Supporting consensus building through argumentative debate is socially important because misunderstanding speaker's intention or emotional conflict sometimes occur among stakeholders. Creation of argumentative corpus and analysis of argumentative discourse is needed for finding the *appropriate* structure of the discourse. The concept of *appropriate* argumentative discourse structure is important for developing argumentation support systems. Often stakeholders need a facilitator to properly organize their ideas. A facilitation system can assist users in building well structured argumentative discussion. To do so, the system must be aware of what kind of argumentative discourse structure is regarded as *appropriate*.

Our focus is argumentative corpus analysis. In the paper we define argumentative discourse structure as appropriate if it tends to lead to *agreement*. We assume that structure of argumentative discourse can be detected with help of *rhetorical relations* that connect related elements in discourse. We think that specific *agreement-oriented* sequences of rhetorical relations that hold across argumentative discourse elements describe the structure tending to lead to consensus. We create a small argumentative

corpus and use Rhetorical Structure Theory relations to annotate it. We, as well, introduce few novel rhetorical relations that we think reflect in a clearer way speaker's intention within question-answering act.

In our study we use data (web discussions) taken from Wikipedia talk pages. Wikipedia talk page provides space for editors to discuss changes to page's associated article or project page. Participants (article editors) launch discussions on different topics related to the article improvement and aim to come to a common point about the topic. We consider this type of web discussions is suitable for our analysis purpose.

To verify our assumption we calculate prior and posteriori probability of rhetorical relations bigrams and present some analysis results.

2 Designing Tag Set

Argumentative corpus, we build, consists of web discussions, where participants express their ideas and aim to reach a consensus. As mentioned above, we focus on the analysis of such argumentative discourse and try to determine the *appropriateness* of its structure. We define the argumentative discourse structure as appropriate if it tends to lead to agreement and assume that it could be detected through sequences of specific agreement-oriented rhetorical relations that hold across discourse elements. So, our primary task was to define the tag set of rhetorical relations that we were going to use for the argumentative corpus annotation.

A powerful instrument that allows analyzing how consecutive discourse elements are related by a small tag set of rhetorical relations is the Rhetorical Structure Theory (RST) proposed by [Mann and Thompson, 1987]. One of the main advantages of RST is its ability to easy and in comprehensive way describe natural texts' structure through rhetorical relations. There are two important reasons why the theory could be suitable for our research. Firstly, though primarily meant for monologue text analysis, RST can also be applied for conversation analysis, as described in some related works [Daradoumis, 1996; Stent, 2000]. Although not fully, theory allows specifying the intentional structure of the discourse, which is very important part for the consensus building process analysis. On the other hand, rhetorical relations enable us to implement software dealing with such intentional structures. Thus, we decide to apply Rhetorical Structure Theory to our study.

Initially, we tried to annotate our corpus with the tag set of rhetorical relations proposed by RST only. Then, during the annotation process, we found that some additional rhetorical relations tags that are not present in RST are needed. These are rhetorical relations that, for example, connect *question-answer* pairs in argumentative discourse and help to determine the intention of the question, such as *Req_evidence* (require evidence) that helps to understand that *Evidence* should be provided as a response and not an *Example* or *Suggestion*. Or, relation tags, reflecting result of the discussion process, such as agreement and disagreement. That is why besides a number of rhetorical relation tags borrowed from RST, we introduce 3 novel relation tags that help to clarify the question intention. We also borrow 10 rhetorical relation tag concepts from related works about building a corpus in the framework of RST [Marcu, 1999]; semantic authoring [Hashida,2007]; addressing behavior in face-to-face conversation [Jovanovic,2006].

2.1 Tag Set to Define Appropriateness

To annotate our corpus we used a tag set of 27 rhetorical relations, a part of which we define as *argumentation-specific* and present it in Table 1. We think these, so called, argumentation-specific relation tags namely reflect structure of argumentative discussion. We structure the tag set as hierarchy of levels and sublevels of rhetorical relations. The uppermost level contains such relations as: *Requirement, Response, Action Request and Politeness*.

- *Requirement* rhetorical relation tags define question intention:
 - *Req_yes/no* tag is used to label the relation between question and statement when confirmation or negation of previously stated information is required
 - *Req_detail* tag is used to label relation between question and statement when additional information (*Example, Explanation, Background, etc.*) is required
 - *Req_evidence* tag is used to label relation between question and statement when evidence is required.
- *Response* rhetorical relation tags define answer intention. Response subsumes the following sublevels of relations *Answer, Argumentation* and *Consensus*:
 - *Answer* contains rhetorical relation tags *Affirmation* and *Negation* used to label the relation between the response and the question, when information is confirmed or negated. It is meant to label the relation between the response and yes/no question
 - *Argumentation* contains rhetorical relation tags *Evidence, Example, Background* and *Explanation_Argumentative*. These tags are used to label the relation between the response and the question, when different kind of argumentation is provided.
 - *Consensus* contains rhetorical relation tags *Agreement* and *Disagreement* that are used to label related discourse elements when agreement or disagreement is expressed by participants.
- *Action Request* rhetorical relation tags define the requirement of performance of an action such as *Request_to_do* and *Suggestion*.
- *Politeness* defines rhetorical relation tags that define appreciation of or regret about an action such as *Gratitude* and *Apology*.

2.2 Elementary Unit

Since we decided to use rhetorical relations to annotate argumentative corpus, one of the issues we tried to deal with was to determine elementary discourse units (text spans) that hold rhetorical relations in our corpus. The simplest solution to this issue was to annotate relations that hold across the *comments* in the discussion. On the other hand, this solution being simple and convenient doesn't fully reflect the structure of the discussions we analyzed. It is obvious that a considerable number of comments contain more than one part that we call here "speech acts". Speech act is a term that refers to the act of successful communicating an intended understanding to the listener. Each "speech act" within one comment has a separate "speech function" like asking question, explaining, etc. One "speech act" can be related to one or more other "speech acts" or comments.

Table 1. Argumentation-specific rhetorical relations tag set which is a part of annotation tag set

| Level | Sublevel | Tag Name |
|----------------|---------------|---------------------------|
| Requirement | | Req_evidence |
| | | Req_detail |
| | | Req_yes/no |
| Response | Answer | Affirmation |
| | | Negation |
| | Argumentation | Evidence |
| | | Explanation_argumentative |
| | | Example |
| | | Background |
| | Consensus | Agreement |
| | | Disagreement |
| Action request | Request_to_do | |
| | Suggestion | |
| Politeness | Gratitude | |
| | Apology | |

3 Data Analysis

The data we selected for our small argumentative corpus are taken from Wikipedia, free encyclopedia Talk pages. The purpose of Wikipedia *talk page* is to provide space for editors to discuss changes to *its* associated article or project page [Wikipedia, free encyclopedia]. For convenience we selected English language pages. We analyzed discussions provided by Moldova talk page¹ and America talk page². We gathered a small corpus containing 693 comments with the total number of participants 197 people. We annotated our data with the tag set containing 27 rhetorical relations, part of which is presented in Table 1. As a result, our corpus includes 627 relations that connect participants' comments. The most frequent relations are listed in Table 2. Basing on frequency results, we can assume that in the type of argumentative discourse we analyzed, rhetorical relations as *Explanation_argumentative*, *Evidence*, *Suggestion*, *Req_evidence* prevail.

Data taken from Wikipedia Talk pages reflect short and long discussions about Wikipedia article editing process. Participants (editors) provide their ideas, suggestions or requests related to the article's improvement. The purpose of the discussions held on Wikipedia talk page is to come to a common point about a certain topic being discussed. To investigate the relationship between consensus building and appropriateness of structure, we count frequencies of bigrams of rhetorical relations (r_1, r_2) , where let r_1 be a preceding relation and r_2 be a succeeding relation that follows r_1 . N-gram of rhetorical relation is determined as shown in Figure 1. Circles present discourse elements (participants' comments) and arcs show how related discourse elements are connected by rhetorical relation. For example, circle *c* has multiple arcs targeted to *a* and *b*, where relations are respectively *Evidence* and *Agreement*.

¹ <http://en.wikipedia.org/wiki/Talk:Moldova>

² http://en.wikipedia.org/wiki/Talk:United_States/Archive

Table 2. Frequent rhetorical relation

| Relation | Frequency | Percentage |
|----------------------------|------------|-------------|
| Explanation_argumentative | 115 | 18% |
| Agreement | 108 | 17% |
| Disagreement | 94 | 15% |
| Evidence | 67 | 11% |
| Suggestion | 49 | 7.8% |
| Justification | 33 | 5.3% |
| Req_evidence | 26 | 4.2% |
| Request_to_do | 22 | 3.5% |
| Req_detail | 19 | 3.0% |
| Affirmation | 10 | 1.6% |
| Other rhetorical relations | 32 | 5.1% |
| Total | 627 | 100% |

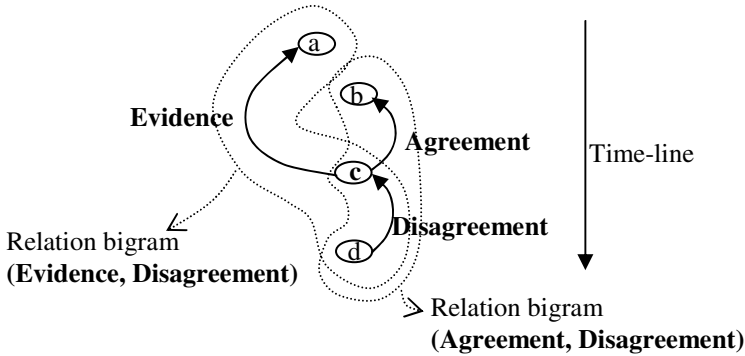


Fig. 1. Determination of bigram of rhetorical relation

In this case, not only (*Agreement, Disagreement*) but also (*Evidence, Disagreement*) are regarded as relation bigrams. *N*-gram of relation, for any *N*, is determined in the same manner. Frequent bigrams and trigrams of rhetorical relations are respectively listed in Tables 3 and 4. It can be seen that most frequently are met the bigrams and trigrams that include *Disagreement* and *Agreement*.

To verify our assumption that there exist specific agreement-oriented sequences of rhetorical relations within argumentative discourse, we calculate priori probability $P(r_2|r_1)$ and posteriori probability $P(r_1|r_2)$, which are respectively defined as

$$P(r_2 | r_1) = \frac{C(r_1, r_2)}{C(r_1)}, \tag{1}$$

$$P(r_1 | r_2) = \frac{C(r_1, r_2)}{C(r_2)}, \tag{2}$$

Table 3. Frequent bigrams of rhetorical relation

| Relation bigram | | Frequency | Percentage |
|------------------------------------|-------------------------------|------------|-------------|
| r_1 | r_2 | | |
| Suggestion | Disagreement | 27 | 6.8% |
| Evidence | Agreement | 16 | 4.0% |
| Agreement | Agreement | 16 | 4.0% |
| Evidence | Disagreement | 14 | 3.5% |
| Disagreement | Agreement | 13 | 3.3% |
| Explanation_ Argumentative | Explanation_ Argumentative | 13 | 3.3% |
| Explanation_ Argumentative | Agreement | 12 | 3.0% |
| Suggestion | Agreement | 11 | 2.8% |
| Disagreement | Explanation_ Argumentative | 11 | 2.8% |
| Disagreement | Disagreement | 9 | 2.3% |
| Other rhetorical relations bigrams | | 245 | 62% |
| Total | | 396 | 100% |

Table 4. Frequent trigram of rhetorical relation

| Relation trigram | | | Frequency | Percentage |
|-------------------------------------|-------------------------------|-------------------------------|------------|-------------|
| r_1 | r_2 | r_3 | | |
| Agreement | Agreement | Agreement | 4 | 1.7% |
| Disagreement | Evidence | Agreement | 3 | 1.3% |
| Explanation_ argumentative | Evidence | Agreement | 3 | 1.3% |
| Explanation_ argumentative | Explanation_ Argumentative | Explanation_ argumentative | 3 | 1.3% |
| Evidence | Disagreement | Evidence | 3 | 1.3% |
| Agreement | Suggestion | Req_detail | 3 | 1.3% |
| Agreement | Suggestion | Agreement | 3 | 1.3% |
| Disagreement | Evidence | Disagreement | 3 | 1.3% |
| Suggestion | Disagreement | Explanation_ argumentative | 3 | 1.3% |
| Explanation_ argumentative | Explanation_ Argumentative | Agreement | 3 | 1.3% |
| Other rhetorical relations trigrams | | | 199 | 83% |
| Total | | | 230 | 100% |

where $C(r)$ and $C(r_1, r_2)$ denote frequencies of a rhetorical relation r and relation bigram (r_1, r_2) , respectively.

These calculations allow us to see which rhetorical relations precede *Agreement* and *Disagreement* rhetorical relations. In Tables 5 and 6 we present some results for agreement and disagreement pairs. Order of relation r_i in the tables is sorted by

$P(r_1|r_2= \text{Agreement})$, the posteriori probability of r_1 when $r_2=\text{Agreement}$, because this probability can be regarded as a contribution of r_1 for building consensus.

Table 5. Priori and posteriori probability for agreement pairs

| Relation r_1 | $P(r_2=\text{Agreement} r_1)$ | | $P(r_1 r_2= \text{Agreement})$ | |
|--------------------------|-------------------------------|-----------|--------------------------------|-----------|
| Evidence | 0.24 | (16 /67) | 0.15 | (16/108) |
| Agreement | 0.15 | (16 /108) | 0.15 | (16/108) |
| Disagreement | 0.14 | (13/94) | 0.12 | (13/108) |
| Explanation_Argumenative | 0.10 | (12 /115) | 0.11 | (12/108) |
| Suggestion | 0.22 | (11/49) | 0.10 | (11/108) |
| Req_evidence | 0.12 | (3/26) | 0.028 | (3/108) |
| Req_detail | 0.11 | (2/19) | 0.019 | (2/108) |
| Request_to_do | 0.09 | (2 /22) | 0.019 | (2/108) |
| Affirmation | 0.1 | (1/10) | 0.0093 | (1/108) |

Table 6. Priori and posteriori probability for disagreement pairs

| Relation r_1 | $P(r_2=\text{Disagreement} r_1)$ | | $P(r_1 r_2=\text{Disagreement})$ | |
|--------------------------|----------------------------------|----------|----------------------------------|---------|
| Evidence | 0.21 | (14 /67) | 0.15 | (14/94) |
| Agreement | 0.037 | (4/108) | 0.043 | (4/94) |
| Disagreement | 0.10 | (9/94) | 0.10 | (9/94) |
| Explanation_Argumenative | 0.043 | (5/115) | 0.10 | (5/94) |
| Suggestion | 0.60 | (27/49) | 0.30 | (27/94) |
| Req_evidence | 0 | (0/26) | 0 | (0/94) |
| Req_detail | 0 | (0/19) | 0 | (0/94) |
| Request_to_do | 0.045 | (1/22) | 0.011 | (1/94) |
| Affirmation | 0 | (0/108) | 0 | (0/108) |

4 Discussion

Creation of argumentative corpus and analysis of *appropriate* argumentative discourse structure is crucial for developing argumentation support systems that assist users in consensus building process. In the paper we assume that *appropriateness* of the discourse structure can be determined through specific rhetorical relations sequences that lead to common agreement. We create a small corpus that consists of web discussions taken from Wikipedia talk page. We call it *argumentative* corpus. We annotate our corpus with rhetorical relations and basing on the annotation, try to analyze discourse structure for *appropriateness*. Namely, we calculate two types of probability, prior and posteriori, for the rhetorical relations bigrams met in our corpus.

Our analysis results are presented in Tables 5 and 6. We sorted data by posteriori probability of *preceding* relation when the *following* relation is *Agreement*, because it can be regarded as a contribution of preceding rhetorical relation for consensus building. The results show that, most frequently, *Agreement* relation are preceded by

Evidence. This is expected, since *Evidence* can be regarded as well measured argumentation support, necessary for consensus. Also, our results might support the idea of importance of clear understanding of the utterance's intention. When, for example, *Evidence* is required and *Example* or *Background* is provided as a response, the information might be insufficient and agreement impossible. Probably, that is why, *Example - Agreement* or *Background - Agreement* pairs are so rarely met. On the other hand, *Explanation_argumentative - Agreement* pair, although, met less frequently than *Agreement - Agreement*, serves to support the assumption that appropriate structure of argumentative discourse could be of several types.

An interesting case is position of *Suggestion* relation frequently followed by both *Agreement* and *Disagreement* rhetorical relations. Still, according to the results *Suggestion-Disagreement* pair prevails. This might be explained by so called *emotional conflict* that often occurs during discussion. *Suggestion* involves requirement for changing hearer's existent belief, which often becomes an obstacle for consensus building process.

It is to be mentioned that although we try to calculate probability for bigrams and trigrams of rhetorical relations, at this stage, in the result analysis we base on the bigrams only. Even though we found few interesting trigram cases that could support our assumption in future, for example, *Req_evidence/Evidence* or *Req_detail/Evidence* pairs tend to more frequently precede *Agreement* relation, our initial data are not sufficient for more reliable verification of existence of longer agreement-oriented patterns in argumentative discourse. Issue that arises here is how many structures leading to agreement could exist. Also, existence of exhaustive argumentation-specific tag set of rhetorical relations is crucial for usability of argumentation support system. Thus another issue that we have to deal with is how to determine a small limited tag set of rhetorical relations suitable for annotation purpose and that will reflect in a proper way argumentative discourse structure.

5 Related Works

Supporting public debate and facilitating consensus building process is actual problem that interests numerous researches. Computer argumentation support systems can assist participants during the multiparty discussions. Most successful approaches in argumentation support systems development base on providing visual "box and arrows" diagramming of argumentation and the analysis of argument constituents. [Reed and Rowe, 2004; Reed and Grasso, 2007; Verheij, 2001]. We think that encouraging user to consider the intention of his next utterance by providing a facilitation function as a list of possible candidates of rhetorical relations would be even more effective for consensus building process support. We assume that there exist appropriate structures (one or more) of argumentative discourse that lead to agreement and we think that these structures could be detected with the help of Rhetorical Structure Theory relations. First step in this approach is building and analyzing an argumentative corpus. In [Mochales and Ieven, 2009], argumentation corpus of legal documents is built and structure analysis is performed argumentation by application of well-known argumentation theories as the background framework for annotation process. We perform our analysis on the basis of web-discussions, assuming that they

reflect common, not strictly logical flow of consensus building conversation. We connect participants' comments within a discussion with RST rhetorical relations. It also must be mentioned that RST application for conversation analysis is not novel. In [Daradoumis, 1996] extended Dialogic RST with new rhetorical relations is used to analyze conversation. The analysis focus is on interruptions in a dialogue (tutorial dialogues are analyzed). In [Stent, 2000] in preliminary result for annotating task-oriented dialogue corpus with RST relations, a new relation Question-Answer is proposed for adjacency pairs. However, we try to design a set of argumentation-specific rhetorical relations that also reflect in a clearer way intention of participant's next utterance.

6 Conclusion

A facilitation system can assist users in building well structured argumentative discussion that would end in common agreement. The concept of appropriate structure of argumentative discourse is significantly important for developing argumentation support computer systems. We designed and built a small corpus to investigate what kind of argumentative discourse structure is appropriate and is leading to consensus building.

Our corpus contains web discussions taken from Wikipedia, free encyclopedia Talk pages. For convenience we selected English language pages. We analyzed discussions provided by Moldova talk page and America talk page. We gathered a corpus containing 693 comments with the total number of participants 197 people and annotated the data with the tag set containing RST rhetorical relations, few relations borrowed from related works and few novel relations we introduced for the research purpose. As a result, our corpus includes 627 relations that connect the comments.

We analyzed the corpus for verifying the assumption that there exist patterns of agreement-oriented sequences of rhetorical relations that tend to lead to consensus within a discussion.

Some preliminary results on rhetorical relations bigrams probability show that bigrams, certain rhetorical relations like *Evidence* or *Explanation_argumentative* will more frequently precede *Agreement* relation. To verify the assumption on longer structure sequences, we firstly need to considerably increase our corpus. We also think that it is important to introduce such parameter as participants ID of comments and consider relationship between participants during the analysis, because these are important factors for facilitating consensus.

For further corpus analysis, we will use the data obtained with help of the computer argumentation support system we are developing. The analysis results, on their turn will be used to improve the facilitation function of the system.

References

1. Carlson, L., Marcu, D., Okurowski, M.E.: Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In: van Kuppevelt, J., Smith, R. (eds.) Current Directions in Discourse and Dialogue, pp. 85–112. Kluwer Academic Publishers, Dordrecht (2003)

2. Mann, W.C., Thompson, S.A.: *Rhetorical Structure Theory: A Theory of Text Organization*, Reprinted from the *Structure of Discourse* (1987)
3. Taboada, M., Mann, W.C.: Applications of Rhetorical Structure Theory. *Discourse Studies* 8(4), 567–588 (2005)
4. Daradoumis, T.: Towards a Representation of the Rhetorical structure of Interrupted Exchanges. In: *Trends in Natural Language Generation: An Artificial Intelligence Perspective*, pp. 106–124. Springer, Berlin (1996)
5. Stent, A.: Rhetorical structure in dialog. In: *Proceedings of First International Conference on Natural Language Generation (INLG 2000)*, pp. 247–252. Mitzpe Ramon, Israel (2000)
6. Reed, C., Rowe, G.: *Araucaria: Software for Argument Analysis, Diagramming and Representation* (2004)
7. Reed, C., Grasso, F.: Recent Advances in Computational Models of Natural Argument. Wiley Periodicals, Inc. *Int. J. Int. Syst.* 22, 1–15 (2007)
8. Verheij, B.: *Artificial Argument Assistants for Defeasible Argumentation*. Elsevier B.V, Amsterdam (2001)
9. Moore, J.D., Paris, C.L.: Planning text for advisory dialogues: Capturing Intentional and Rhetorical Information. In: *Computational linguistics - Association for Computational Linguistics*, pp. 651–694. MIT Press, Cambridge (1993)
10. Jovanovic, N., den Akker, R.o., Nijholt, A.: *A Corpus for Studying Addressing Behaviour in Multi-Party Dialogues*. Springer Science + Business Media B.V, Heidelberg (2006)
11. Hashida, K.: Semantic Authoring and Semantic Computing. In: Sakurai, A., Hasida, K., Nitta, K. (eds.) *JSAI 2003. LNCS (LNAI)*, vol. 3609, pp. 137–149. Springer, Heidelberg (2007)
12. Mochales, R., Ieven, A.: Creating an Argumentation Corpus: Do Theories Apply to Real Arguments? In: *ICALL-2009, Barcelona, Spain* (2009)

Improving Identification Accuracy by Extending Acceptable Utterances in Spoken Dialogue System Using Barge-in Timing

Kyoko Matsuyama, Kazunori Komatani, Toru Takahashi,
Tetsuya Ogata, and Hiroshi G. Okuno

Graduate School of Informatics, Kyoto University, Kyoto, Japan
{matuyama, komatani, tall, ogata, okuno}@kuis.kyoto-u.ac.jp

Abstract. We describe a novel dialogue strategy enabling robust interaction under noisy environments where automatic speech recognition (ASR) results are not necessarily reliable. We have developed a method that exploits utterance timing together with ASR results to interpret user intention, that is, to identify one item that a user wants to indicate from system enumeration. The timing of utterances containing referential expressions is approximated by Gamma distribution, which is integrated with ASR results by expressing both of them as probabilities. In this paper, we improve the identification accuracy by extending the method. First, we enable interpretation of utterances including ordinal numbers, which appear several times in our data collected from users. Then we use proper acoustic models and parameters, improving the identification accuracy by 4.0% in total. We also show that Latent Semantic Mapping (LSM) enables more expressions to be handled in our framework.

Index Terms: spoken dialogue systems, conversational interaction, barge-in, utterance timing.

1 Introduction

Natural conversational dialogue systems should allow users to freely express their utterances anytime. Of particular importance is that the user should be able to interrupt the system's utterances. This ability to **barge in** is useful to convey the user's intention. The user should be able to occasionally interrupt the system by specifying an item when the system is listing items. For example, the system and the user can interact as follows:

User. Tell me which temple you suggest visiting.

System. There are ten temples that I would suggest. "Kinkaku-ji Temple", "Ginkaku-ji Temple..."

User. That one.

System. OK, you mean "Ginkaku-ji temple." It is the most famous one ...

In this case, the user interrupts the system while it reads out "Ginkaku-ji temple." This system identifies the user's referent, that is, what the user indicates by

“That one.” By using the barge-in timing of the user utterance, it determines that “Ginkaku-ji Temple” is specified by the user. This kind of dialogue in which items are read out in a list is important for two reasons. First, the user can indicate the referent by timing information, which is detected robustly. Barge-in timing is more reliable than ASR results in many cases. Therefore, this new dialogue strategy enables the system to obtain the user intention by reading out each item even in noisy environments. Second, this dialogue often appears when a system displays a retrieval result in the information retrieval task. This task is a promising one for conversational dialogue systems and is being developed at several companies such as Microsoft [1] and Google[4].

We have developed a method for identifying the user’s *referent* during system enumeration by focusing on barge-in utterances while the system lists choices [2]. Our purpose is to identify the user’s referent with a high degree of accuracy. We exploit utterance timing together with ASR results to identify the user’s referent as follows. First, we determine the relationships between the timing and content of a user utterance in order to use timing information. Then we construct a framework in which both timing information and ASR results are represented as probabilities. By using these probabilistic representations, we can obtain the most relevant interpretation as the one having the maximum likelihood [2]. We furthermore improve the interpretation obtained from ASR results in order to handle user utterances that include no content words in each item. Specifically, we introduce the interpretation of utterances with numbers. We also propose interpreting utterances that include words related to the items. We collect documents from the Web and use Latent Semantic Mapping (LSM) [3] to measure the closeness between the utterance and each item.

Interpretation using utterance timing has not been investigated although barge-in has attracted the attention of researchers concerned with spoken dialogue systems, specifically, the issue of barge-in detection [4,5]. Their purpose has been to detect users’ barge-in occurrences quickly and accurately. McTear [6] focused on how to stop a system utterance in order to recognize a user’s barge-in. Ström [7] discussed a system’s behavior when barge-ins were incorrectly detected. We report a new interpretation that utilizes the locutionary act of barge-in, on the assumption that the barge-in detection is correct.

2 Modeling of User’s Utterance Timing

We investigate the relationships between the content of user utterances and utterance timing to utilize barge-in timing. Here, we define **utterance timing** as the temporal subtraction of when a system utterance starts and when a user utterance starts (see Figure 1). While a system enumerates choices for a selection, the user utters **referential expressions** or **content expressions** to select one item. The former indicates an utterance that contains a reference term, such as “that one” or a pronoun. The latter indicates an utterance containing content words, such as “Kinkaku-ji Temple.” If the user utters a content expression,

¹ <http://www.google.com/goog411/>

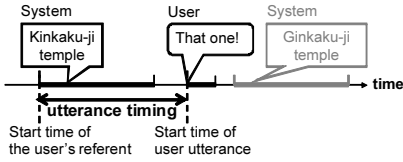


Fig. 1. Definition of utterance timing

Table 1. Two different conditions

| Condition | Cond. A | Cond. B |
|-------------------|---------|---------|
| # user utterances | 35 | 69 |
| PAUSE (sec.) | 1.0 | 2.0 |
| AVERAGE (sec.) | 0.73 | 5.27 |

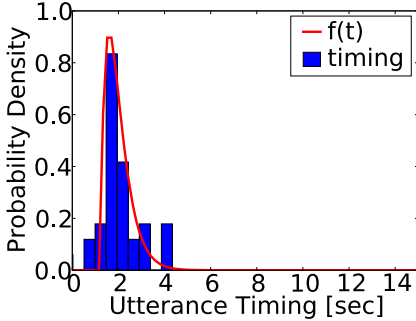


Fig. 2. Timing distribution in Cond. A

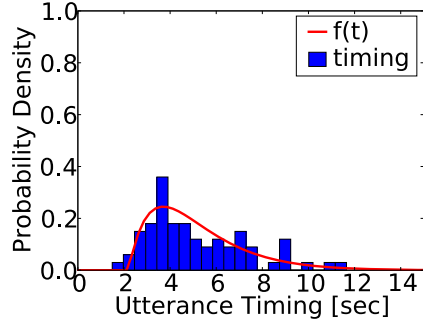


Fig. 3. Timing distribution in Cond. B

the user conveys his intention not by the timing but by the content. On the other hand, a characteristic distribution of the utterance timing must be in the referential expression to convey a user's intention.

We determine how utterance timing of referential expressions is distributed. We collected user utterances under two different conditions (see Table 1). PAUSE represents the interval of time between items and AVERAGE represents an average length of enumerated items. Utterance timing is detected by using the voice activity detection of an ASR engine, Julius [8]. The distributions of utterance timing of both conditions are shown in Figures 2 and 3 as histograms. The bars in the histograms denote the relative frequencies of utterances in their timing, multiplied by the bar's width to represent the probabilistic density. The widths are set to 0.5 seconds. We can see clear peaks in both figures, although their peak positions and attenuation are different.

We model the histograms representing utterance timing of referential expressions by Gamma distribution:

$$f(t) = \frac{1}{(\sigma - 1)! \rho^\sigma} (t - \mu)^{\sigma-1} e^{-(t-\mu)/\rho} \quad (1)$$

Zhou *et al.* also claimed that the time required for human perception follows Gamma distribution [9]. Equation (1) has three parameters: μ , ρ , and σ . The details of how these parameters are set was explained in our previous paper [2]. The Gamma distributions are also illustrated in Figures 2 and 3. Their parameters are as follows: $\mu = 1.2$, $\rho = 0.3$ and $\sigma = 2.0$ in Figure 2; $\mu = 2.2$, $\rho = 1.5$ and $\sigma = 2.0$ in Figure 3.

3 Identifying User's Referent Using Barge-in Timing and ASR Results

We present a framework in which both utterance timing and ASR results are uniformly represented as probabilities. This enables us to identify a user's referent as an item having the maximum likelihood.

3.1 Basic Formulation

We formulate the problem of identifying a user's referent by calculating T_i such that the probability $P(T_i|U)$ is maximized. Here, T_i denotes the i -th item enumerated by a system, and U denotes a user utterance. That is, $P(T_i|U)$ represents how probable it is that U indicates T_i corresponding to each item in the system's enumeration. We calculate the probability for each T_i and then determine the user's intention, T .

$$T = \operatorname{argmax}_{T_i} P(T_i|U) = \operatorname{argmax}_{T_i} \frac{P(U|T_i)P(T_i)}{P(U)} = \operatorname{argmax}_{T_i} P(U|T_i) \quad (2)$$

We assume all the prior probabilities $P(T_i)$ are equal. $P(U)$ is not dependent on i .

We calculate $P(U|T_i)$ in accordance with Equation (2) by considering the possibilities of two cases: interpreting user's intention by either the utterance timing, C_1 or the content of the utterance, C_2 . Thus, $P(U|T_i)$ can be represented as the following sum:

$$P(U|T_i) = \sum_{k=1,2} P(U|T_i, C_k)P(C_k|T_i) \quad (3)$$

$$= \frac{1}{2} \sum_{k=1,2} P(U|T_i, C_k) \quad (4)$$

Here we assume that these prior probabilities $P(C_k|T_i)$ are even. We set the coefficient α as the score ranges between $P(U|T_i, C_1)$ and $P(U|T_i, C_2)$ by setting a parameter α , as shown in Equation (5).

$$P(U|T_i) = (1 - \alpha)P(U|T_i, C_1) + \alpha P(U|T_i, C_2) \quad (5)$$

Equation (5) denotes that the two cases are considered for all user utterances. $P(U|T_i, C_k)$ denotes the probability of an occurrence of user utterance U in the case of C_k for each item T_i . We assume that U contains two elements: $U = \{X, t_b\}$. Here, X indicates an ASR result and t_b denotes the time at which the user barges in during the system's utterance. Both $P(U|T_i, C_1)$ and $P(U|T_i, C_2)$ are defined in the following subsections. The flow of our method of identifying a user's referent is shown in Figure 4.

3.2 Probability Defined by Using Barge-in Timing

We define $P(U|T_i, C_1)$ by using utterance timing since C_1 is defined as the case when a user expresses his intention by using utterance timing. Therefore, we

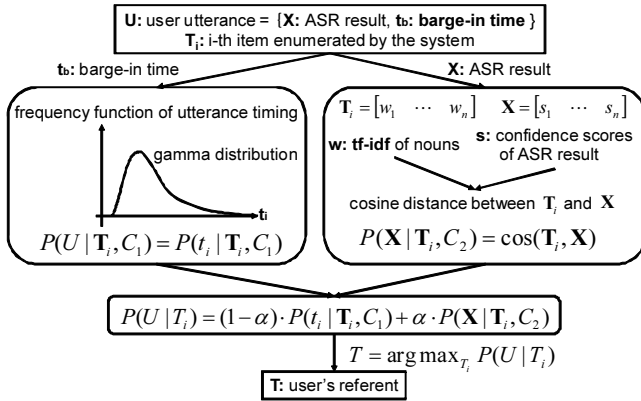


Fig. 4. Flow of identifying user's referent

assume probability $P(U|T_i, C_1)$ depends not on an ASR result X but on barge-in time t_b only. Here, t_i denotes the utterance timing after the system starts enumerating item T_i (see Figure 1); that is,

$$t_i = t_b - start(T_i) \tag{6}$$

Thus, $P(U|T_i, C_1)$ is calculated as follows:

$$P(U|T_i, C_1) = P(t_i|T_i, C_1) \tag{7}$$

Note that the probability $P(t_i|T_i, C_1)$ represents a case when a user indicates a specific item, T_i , in timing t_i . Therefore, the probability corresponds to the Gamma distribution we found in Section 2. We use the distribution $f(t_i)$ as $P(t_i|T_i, C_1)$.

3.3 Probability Defined by Using ASR Results

The probability $P(U|T_i, C_2)$ represents how close a user utterance U (ASR result X) and each item T_i are. We define $P(U|T_i, C_2)$ by using an ASR result in accordance with the definition of C_2 [2], except for some utterances for which we also need to use barge-in timing t_b . One example utterance is “The item before last.” This example needs to be interpreted by using both the user’s barge-in timing and the ASR result. That is, we need to know what a user said, and when.

The closeness is defined by cosine distance:

$$P(U|T_i, C_2) = cos(\mathbf{T}_i, \mathbf{X}) \tag{8}$$

where \mathbf{X} and \mathbf{T}_i are M -dimensional vectors. M is the vocabulary size of the system. The elements of \mathbf{T}_i are TF-IDF values [10] of all nouns in the enumerated items in order to account for the word importance. The vector \mathbf{X} corresponds to

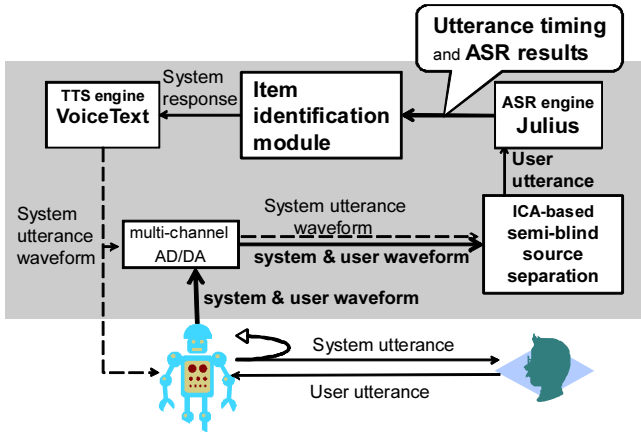


Fig. 5. System architecture

the ASR result for the user utterance U . This vector consists of ASR confidence scores for the M nouns. By considering ASR confidence scores when calculating the probability, damage caused by ASR errors is alleviated.

To interpret utterances that include numbers such as “The second one”, we add such number words into the vocabulary. For example, “first” is added to the vector T_1 corresponding to the first item. The size of X also increases accordingly. After adding these words, “the second” can be interpreted to indicate the second item in the system’s enumeration, for example. When “The item before last” is recognized, we first estimate the interrupted item by using barge-in timing t_b and calculate the most likely user selection by the number of the items and ASR result. Then we assign the average confidence scores for words in ASR results to the corresponding element of vector X .

4 Experimental Evaluation

4.1 Implementation of Barge-in-able Dialogue System

The overview of the architecture of our barge-in-able dialogue system is depicted in Figure 5. The process flow is summarized as follows: The multi-channel AD/DA, RASP of JOEL System Technology captures a mixed sound and the wave file of the system utterance into a 2-channel wave stream. The ICA-based semi-blind source separation [11] obtains this wave stream and separates the user utterance incrementally. The ASR engine, Julius [8], then recognizes the separated user utterance and begins to record when the utterance starts. The item-identification sub-system identifies the user’s referent on the basis of ASR results and the barge-in timing and generates a system response. We used VoiceText² developed by PENTAX Inc. as a Text-to-Speech (TTS) engine.

² <http://voice.pentax.jp/>

4.2 Conditions of Experimental Evaluation

We collected 400 utterances from 20 subjects. The utterances consisted of 263 referential expressions and 137 content expressions. The system listed news titles in 10 RSS feeds, and the subjects were told they could interrupt the system utterance and say whatever they liked. The number and length of titles are different for each RSS feed. We set three pause lengths between enumerated items: 1.5, 2.0, and 3.0 seconds. The parameters of the Gamma distribution used in our method were determined beforehand as follows: $\mu = 0.73$ and $\sigma = 2.0$. The parameter ρ of Gamma distribution was determined in accordance with the pause lengths between items and the contents of enumerated items. We set α in Equation (5) to 0.6 empirically. Accuracies when α is changed are shown in Section 4.3. We used an acoustic model containing pink noise, which reflects the actual acoustic environment. We made a statistical language model by using the CIAIR corpus [12] and news articles obtained from each RSS feed. On average, the vocabulary size was 5835.

We evaluated several methods by identification accuracies, that is, how well the system correctly identified the user's referents. Each method is listed below:

Cond. 1: Use of barge-in timing only

A user's referent was the item that had just been read out or presented when a user started speaking.

Cond. 2: Use of barge-in timing model only

A user's referent was identified by the using the timing model of Gamma distribution.

Cond. 3: Our method (not extended to interpret numbers)

A user's referent was identified by the identical method to [2].

Cond. 4: Our method (explained in Section 3)

A user's referent was identified by our method extended to interpret numbers.

Conds. 1 and 2 correspond to simpler methods in which no ASR results are used. We set these to verify how well the timing model works and whether ASR results are necessary or not. In Cond. 3, the vector size M and the number of items N varied with the number of enumerated news articles. On average, M was 104.5, and N was 15.8. In Cond. 4, M was 173.5. The ASR word accuracy for all utterances was 38.3%. Reasons for the low accuracy include sound reflections or distortions during the sound source separation since we used a microphone embedded in a robot instead of using a normal close-talk microphone. Also, correctly recognizing a user's utterances is difficult because these users often speak quickly or quietly.

4.3 Experimental Results

The identification accuracies of the user's referent for 263 utterances with referential expressions, 137 utterances with content expressions, and all 400 utterances are shown in Table 2. Accuracy of Cond. 2 was better than that of Cond. 1.

Table 2. Identification accuracy [%] for user utterances

| Condition | Referential expression (#:263) | Content expression (#:137) | Total (#:400) |
|-------------------------------|-----------------------------------|-------------------------------|------------------|
| 1: only barge-in timing | 84.8 | 25.5 | 64.5 |
| 2: only barge-in timing model | 87.8 | 32.1 | 68.8 |
| 3: our method | 81.4 | 53.3 | 71.8 |
| 4: + numbers | 85.2 | 57.7 | 75.8 |

Table 3. Identification accuracy [%] for α in Cond. 4

| α value | Referential expression (#:263) | Content expression (#:137) | Total (#:400) |
|----------------|-----------------------------------|-------------------------------|------------------|
| 0.0 | 87.8 | 32.1 | 68.8 |
| 0.2 | 86.7 | 42.3 | 71.5 |
| 0.4 | 85.9 | 54.7 | 75.3 |
| 0.6 | 85.2 | 57.7 | 75.8 |
| 0.8 | 84.8 | 56.9 | 75.3 |
| 1.0 | 0.76 | 43.1 | 15.3 |

This result shows the utterance timing model formulated as Gamma distribution works effectively. Moreover, the timing information is also effectively used for interpreting content expressions, because some content utterances were identified correctly even though users conveyed their referent by content words.

The identification accuracy of Cond. 3 was 71.8% for all utterances, outperforming the accuracies of Conds. 1 and 2. In particular, the accuracy for content expressions also improved by 21.2 points compared with that of Cond. 2. The result suggests using the ASR results is effective although its accuracy is not high. The identification accuracy of Cond. 4 was 75.8% for all utterances, which outperformed the accuracy of Cond. 3. In fact, the identification accuracy of content expressions including numbers improved by 27 points more than that of Cond. 3. The differences between Cond. 3 and 4 for referential expressions and total utterances were statistically significant ($p < 0.01$) by t-tests. Most significantly, the accuracy for referential expressions of Cond. 4 also improved by 3.8 points more than that of Cond. 3. These utterances can be identified after scores of incorrect ASR results decreased due to the number being considered.

The highest accuracy of referential expressions was obtained by Cond. 2. This case corresponds to $\alpha = 0.0$. Table 3 lists identification accuracies in Cond. 4 when α is changed from 0.0 to 1.0. When we set α to 1.0, a user’s referent is identified by only $P(U|T_i, C_2)$. In this case, the identification accuracy of referential expressions is very low because ASR results of referential expressions such as “That one” contain no information associated with any items. When we set α smaller, $P(U|T_i, C_1)$ was emphasized and more referential expressions were correctly identified. This result indicates the trade-off between $P(U|T_i, C_1)$ and $P(U|T_i, C_2)$. To improve the accuracy for referential expressions in our methods, we should dynamically determine α in Equation (5) for each user’s utterance.

Table 4. Identification accuracy [%] by using LSM

| Condition | Referential expression (#: 263) | Content expression (#: 137) | Total (#: 400) |
|-----------|------------------------------------|--------------------------------|-------------------|
| Using LSM | 85.2 | 58.3 | 76.0 |

5 Extending Acceptable Utterances by LSM

The user often tries to convey his or her intention using related words, that is, content words that were not included in the enumerated items. This utterance, for instance, includes “The Beckham’s result” corresponding to the item “Soccer.” To deal with this utterance, we collect the documents obtained by copying sentences from Wikipedia³ pages related to each item. Here, $P(U|T_i, C_2)$ represents how close a user utterance U (ASR result X) and the documents from the Web corresponding to each item T_i are, and it is calculated by using LSM [3]. We decompose the co-occurrence matrix to obtain the k -dimensional vectors of all the documents. We construct a $M \times N$ co-occurrence matrix between the items and the documents, where M is the vocabulary size and N is the total number of the documents. We applied singular value decomposition (SVD) to the matrix and compressed its rank to k . Here, k corresponds to $N - 2$. The k -dimensional vectors were calculated on the basis of the matrix obtained from the SVD.

We estimate $P(U|T_i, C_2)$ by calculating the cosine distance between the k -dimensional vectors of the user’s utterance and those of the documents. The user’s utterance was recognized using a statistical language model that was based on the documents for each RSS feed. The documents consist of the data from Wikipedia and the 115 command utterances such as “Let me hear the news”. On average, the size of the vocabulary was 17253. The ASR word accuracy was 37.5%. The size of the co-occurrence matrix M corresponds to the size of vocabulary. The k -dimensional vector of the user’s utterance was calculated from its ASR confidence scores and the matrix obtained from the SVD.

We apply LSM only when a user specifies the item by using related words to avoid misinterpretation by applying LSM to all utterances. We compare two acoustic likelihoods to select utterances to apply LSM. One is calculated by using a language model for LSM and the other by using language model used in Cond. 4. We obtain the difference between them by subtracting the latter from the former. We use LSM only when the difference is more than 90. This value is empirically determined.

We evaluated the effectiveness of using LSM to identify the user’s referent. The identification accuracy by using LSM is shown in Table 4. Here we set α in Equation (5) to 0.6. Table 4 shows that the identification accuracy outperformed that of Cond. 4. In fact, the one utterance that only has a content expression with related words in the data became identified correctly.

³ <http://ja.wikipedia.org/>

6 Conclusion

We created a novel model of users' barge-in timing and developed an identification method by integrating the timing model with ASR results as a probabilistic representation. As a result, we made a barge-in-able conversational dialogue system that reads out news articles obtained from RSS feeds.

Our method covers only a sub-dialogue where a user selects one item when a system lists choices. In a natural conversational interaction, users can make a variety of barge-in utterances; for example, to conclude the conversation quickly, to correct misunderstandings, or to assert themselves strongly - not only to indicate their referent. Nevertheless, this work is the first step towards achieving such an intuitive interaction in conversational dialogue systems. We developed a new interaction exploiting barge-in timing model and showed that it can improve the accuracy of identifying a user's referent, especially in barge-in-able conversational dialogue systems.

References

1. Wang, Y.Y., Yu, D., Ju, Y.C., Acero, A.: An introduction to voice search. *IEEE Signal Processing Magazine* (May 2008)
2. Matsuyama, K., Komatani, K., Ogata, T., Okuno, H.G.: Enabling a User to Specify an Item at Any Time During System Enumeration - Item Identification for Barge-In-Able Conversational Dialogue Systems. In: *Interspeech-2009*, pp. 252-255 (2009)
3. Bellegarda, J.R.: Latent semantic mapping. *IEEE Signal Processing Magazine* 22(5), 70-80 (2005)
4. Rose, R.C., Kim, H.K.: A hybrid barge-in procedure for more reliable turn-taking in human-machine dialogue systems. In: *Proceeding of IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 198-203 (2003)
5. Ljolje, A., Goffin, V.: Discriminative training of multi-state barge-in models. In: *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 353-358 (2007)
6. McTear, M.F.: Spoken Dialogue Technology: Enabling the Conversational User Interface. *ACM Computing Surveys*, 90-169 (2002)
7. Ström, N., Seneff, S.: Intelligent Barge-in in Conversational Systems. In: *Proceeding of International Conference on Spoken Language Processing* (2000)
8. Kawahara, T., Lee, A., Takeda, K., Itou, K., Shikano, K.: Recent progress of open-source LVCSR Engine Julius and Japanese model repository. In: *Proceeding of International Conference on Spoken Language Processing*, pp. 3069-3072 (2004)
9. Zhou, Y., Gao, J., White, K., Merk, I., Yao, K.: Perceptual Dominance Time Distributions in Multistable Visual Perception. *Biological Cybernetics* 90(4), 256-263 (2004)
10. Salton, G.: *Automatic Text Processing*. Addison-Wesley, Reading (1988)
11. Takeda, R., Nakadai, K., Komatani, K., Ogata, T., Okuno, H.G.: Barge-in-able Robot Audition Based on ICA and Missing Feature Theory under Semi-Blind Situation. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1718-1723 (2008)
12. Kawaguchi, N., Matsubara, S., Takeda, K., Itakura, F.: CIAIR In-Car Speech Corpus -Influence of Driving Status-. *IEICE Transactions on Information and Systems*, 578-582 (2005)

A New Approach to Construct Optimal Bow Tie Diagrams for Risk Analysis

Ahmed Badreddine and Nahla Ben Amor

LARODEC, Institut Supérieur de Gestion Tunis, 41 Avenue de la liberté,
2000 Le Bardo, Tunisie

badreddine.ahmed@hotmail.com, nahla.benamor@gmx.fr

Abstract. Bow tie diagrams have become popular methods in risk analysis and safety management. This tool describes graphically, in the same scheme, the whole scenario of an identified risk and its respective preventive and protective barriers. The major problem with bow tie diagrams is that they remain limited by their technical level and by their restriction to the graphical representation of different scenarios without any consideration to the dynamic aspect of real systems. This paper overcomes this weakness by proposing a new Bayesian approach to construct bow ties from real data.

Keywords: Bow tie diagrams, Risk analysis, Bayesian networks, Learning structures, Learning parameters.

1 Introduction

Since 2003, bow tie diagrams [5] have been used as a tool for risk analysis in several industrial fields such as energetic, automobile etc. Their success can be explained by the fact that the whole scenario for each identified risk also called *top event* (TE) is clearly represented via two parts: the first corresponds to a *fault tree* defining all possible causes leading to the TE and the second represents an *event tree* to reach all possible consequences of the TE. In addition, bow tie diagrams allow to define in the same scheme *preventive barriers* to limit the occurrence of the TE and *protective barriers* to reduce the severity of its consequences. In spite, its widely use in many organizations, this method remains limited by its technical level which is restricted to graphical presentation of different scenarios without any suggestion about optimal decisions regarding the expected objectives. In the literature few researches have been carried out to deal with the building phase of bow tie diagrams and their exploitation in the decision problems. Indeed, we have noticed that the researchers are usually interested in its quantification phase [7] [9], while the construction one is always assigned to the experts. We can in particular mention [6] where different steps have been proposed to build the bow tie diagrams, this approach is mainly based on the experts knowledge. This paper proposes a new Bayesian approach to construct bow tie diagrams which reflect the real behavior of the existing systems. In fact, we will generalize the actual *deterministic* bow tie diagrams to probabilistic ones,

by replacing the logical AND and OR gates by conditional probability tables (CPTs). Our approach is based on two phases, namely, a structure learning phase relative to the graphical component of bow ties and a parameters learning phase relative to their numerical component.

The remainder of this paper is organized as follows: Section 2 presents a brief recall on the bow tie diagrams analysis. Section 3 details our new learning approach. And finally, section 4 presents an illustrative example in the petroleum field.

2 A Brief Recall on the Bow Tie Diagrams Analysis

The bow tie diagrams are a very popular and diffused probabilistic technique developed by Petroleum companies for dependability modeling and evaluation of large safety-critical systems [5]. The principle of this technique is to build for each identified risk R_i (also called *top event* (TE)) a bow tie representing its whole scenario on the basis of two parts, as shown in figure 1:

- the first part corresponds to the left part of the scheme which represents a *fault tree* (FT) defining all possible causes leading to the (TE). These causes can be classified into two kinds: The first are the initiator events (IE) which are the principal causes of the TE , and the second are the undesired or critical events ($IndE$ and CE) which are the causes of the IE . The construction of the left part proceeds in top down manner (from TE to $IndE$ and CE). The relationships between events and causes are represented by means of logical AND and OR gates.
- The second part corresponds to the right part of the scheme which represents an *event tree* (ET) to reach all possible consequences of the TE . These consequences can be classified into three kinds: *second events* (SE) which are the principal consequences of the TE , *dangerous effects* (DE) which are the dangerous consequences of the SE and finally *major events* (ME) of each DE . The construction of the event tree proceeds as the fault tree i.e. in top down manner.

The bow tie diagrams also allows the definition, in the same scheme, of some *preventive barriers* to limit the occurrence of TE and also of *protective barriers* to reduce the severity of its consequences. These barriers can be classified as *active* if they require a source of energy or a request (automatic or manual action) to fulfill their function (e.g. a safety valve, an alarm etc.) or *passive* if they do not need a source of energy nor a request to fulfill their function (e.g. a procedure, a retention dike, a firewall etc.).

The major problem with bow tie diagrams is that they are limited by their technical level and by their restriction to a graphical representation of different scenarios without any consideration to the dynamic aspect of the real systems. In fact, the logical AND and OR gates represent deterministic causal relationships, which do not always reflect the real systems behavior. For instance, the OR gates means that the fail of a component implies the global fail of the related

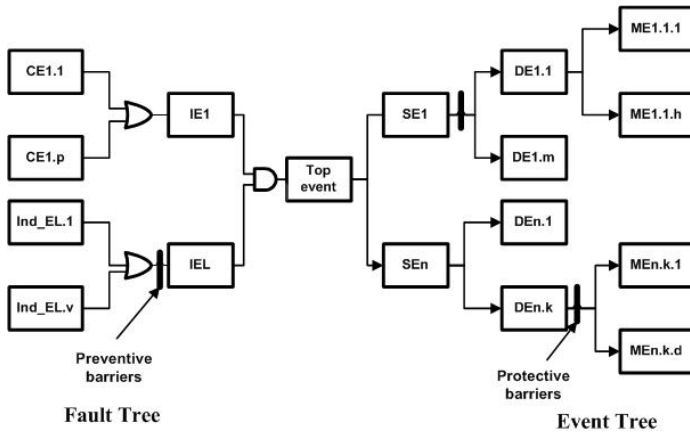


Fig. 1. A bow tie diagram model

system while we can easily imagine that some functionalities of this system will be maintained even with a small probability. In addition, The choice of the appropriate barriers is not an easy task, since it depends on many criteria such as effectiveness, reliability, availability and cost [5]. Thus their definition from expert experience without any consideration of real data may affect their quality since it seems unrealistic to suggest static recommendations in real dynamic systems. To overcome this problem we propose to learn bow ties from real data and to improve them by adding a new numerical component allowing us to model in a more realistic manner the system behavior.

3 A New Algorithm to Construct Bow Tie Diagrams

Our aim is to construct bow ties which reflect the real behavior of the existing systems i.e which are not exclusively based on the experts knowledge in order to make them more effective and useful. To this end we will consider bow ties as probabilistic graphs, denoted by *BT*, having:

- a tree-structured graphical component *T* on a set of *n* nodes $V = \{X_1, \dots, X_n\}$ s.t. each node X_i represents an event (e.g. *IE*, *CE*, *SE* etc.). All these events are considered as binary (present or absent), thus all variables in *V* can have two states True (T) or False (F). *T* can be dispatched into two subtrees such that *TE* represents the unique root of the first one corresponding to the *ET* and the unique leaf of the second one corresponding to the *FT*. In the remainder X_1 is considered as the top event *TE*. The set of different arcs connecting nodes in *V* is denoted by *A*.
- a numerical component allowing us in one hand to characterize the impact of different causes on the top event *TE* and in the other hand to study its repercussion while considering its severity and those of its consequences. Thus

- to each node X_i in FT , we will assign a conditional probability table (CPT) in the context of its parents (i.e. $P(X_i | Pa(X_i))$ where $Pa(X_i)$ denotes the parent set of (X_i)). These tables define the behavior of different events regarding their causes. This means that they will generalize the logical AND and OR gates. For instance, if X_2 AND X_3 cause X_1 , this means that in the CPT lied to X_1 , $P(X_1 = T | X_2 = T, X_3 = T) = P(X_1 = F | X_2 = T, X_3 = F) = P(X_1 = F | X_2 = F, X_3 = T) = P(X_1 = F | X_2 = F, X_3 = F) = 1$ and that the remaining entries are null. The same relation can be represented with a more flexibility via probability degrees pertaining to the unit interval.
- to each node X_i in ET , we will assign a value relative to its severity w.r.t each of its children X_j (i.e. its consequences). This value quantifies the impact of a realization of X_i on X_j . In the literature several methods are proposed to define the severity of an event. Here we will consider that this value is equal to $P(X_j = T | X_i = T)$.

Roughly speaking, bow ties here will have, almost, the same aspect than *classical* ones, except for AND and OR gates, which will be replaced by CPTs. Moreover they will have an additional component defining severity values in the ET . Regarding the barriers, we propose to define them in a dynamic way as detailed in section 4. Note that probabilistic trees are particular case of Bayesian networks which are powerful tools in reasoning under uncertainty. Thus to build bow ties we can use any standard learning algorithm relative to probabilistic trees as described below.

3.1 New Approach to Learn Bow Ties Structure

To learn bow ties structure, we propose to use the standard tree building algorithm proposed by Chow and Liu [3]. This algorithm is derived from the Maximal Weight Spanning Tree (MWST) [4] and has as input a training set (denoted by TS) which generally corresponds to a set of records observed during a given period, and as output a spanning tree, denoted by $UT = \{U, E\}$ s.t. U is the set of nodes and E is the set of edges (since UT is undirected). This algorithm, has shown its success to obtain the optimum tree structure from training set [8]. In order to find different dependencies, this algorithm uses the mutual information $I_{i,j}$ between each pair of variables (X_i, X_j) s.t. $i \neq j$ in TS expressed as follows:

$$I_{i,j} = \sum_{x_i x_j} P_{ij}(x_i, x_j) \log\left(\frac{P_{ij}(x_i, x_j)}{P_i(x_i)P_j(x_j)}\right) \tag{1}$$

where $P_{ij}(x_i, x_j)$ (resp. $P_i(x_i)$) is the proportion of observations in the training set TS s.t. $X_i = x_i$ and $Y_i = y_i$ (resp. $X_i = x_i$) i.e. the number of these observations divided by the whole number of observations in TS . Formally, Chow and Liu algorithm [3] can be outlined as follows (in this version the root is defined as a prior):

Algorithm 1. Learning_undirected_tree_structure

Data: TS on a set of n variables $N = \{X_1, \dots, X_n\}$; X_1 as root

Result: $UT = \{U, E\}$

begin

```

for  $i \in \{1, \dots, n - 1\}$  do
    for  $j \in \{2, \dots, n\}$  do
        Compute the mutual information  $I_{ij}$  using equation (1)
         $M[i][j] \leftarrow I_{ij}$ 

```

$U \leftarrow X_1, E \leftarrow \emptyset;$

while $|U| < n$ **do**

```

    Use  $M$  to find  $X_i$  in  $N$  and  $X_j$  in  $N - \{U\}$  s.t.  $I_{ij}$  is the highest mutual
    information (within all possible combinations)

```

$U \leftarrow U \cup X_j$

```

     $E \leftarrow E \cup (i - j)$ 

```

end

Thus, our idea is to run the algorithm **1** twice in order to learn the FT and the ET structures (denoted by T_{FT} and T_{ET}) separately from two training sets: the first, denoted by TS_{FT} , is relative to the causes leading to TE and the second, denoted by TS_{ET} , is relative to its consequences. In these two phases TE will be considered as the root. Then we will orient the resulted undirected trees *semantically* using the fact that events in T_{FT} are causes of TE i.e. arcs in T_{FT} will be directed towards TE and events in T_{ET} are its consequences i.e. arcs in T_{ET} will be backwards TE .

3.2 Learning Bow Ties Parameters

Once the bow structure is fixed, we will learn its parameters in order to define its numerical component. As described above, this component differs from FT and ET , thus computations can be done as follows:

To quantify the fault tree FT , we will use a Bayesian approach based on informative priors. More precisely, to estimate $P(X_i = k \mid Pa(X_i) = j)$ (i.e. the probability that X_i is equal to k knowing that its parents denoted by $Pa(X_i)$ take the value j) we will use the maximum a posterior (MAP) estimate expressed by:

$$\hat{P}(X_i = k \mid Pa(X_i) = j) = \frac{N_{ijk} + \alpha_{ijk}}{\sum_k N_{ijk} + \alpha_{ijk}} \tag{2}$$

where N_{ijk} is the number of instances in the training set TS_{FT} where $X_i = k$ and $Pa(X_i) = j$ occur conjointly and α_{ijk} is a Dirichlet prior having a simple interpretation in terms of pseudo counts i.e. we suppose that we saw the value k of X_i for each value j of $Pa(X_i)$ α_{ijk} times. This value prevents us from declaring that the event $(X_i = k, Pa_i = j)$ is impossible just because it was not seen in the training set (which is the case of the standard maximum likelihood (ML)

estimate). Thus if $\alpha_{ijk} > 0$ then $\hat{P}(X_i = k \mid Pa(X_i) = j)$ will not be equal to 0. In what follows, we will use uniform prior i.e. $\forall i, j, k \alpha_{ijk} = 1$.

To compute severity degrees relative to ET , we should compute for each node X_i in ET (except ME), a vector S_i s.t. $S_i[j]$ is the severity of X_i w.r.t to its children X_j . In the literature several methods are proposed to define the severity of an event. Here we will use TS_{ET} to compute it by considering that $S_i[j] = P(X_j = T \mid X_i = T)$. To compute this value we use Bayes theorem as follows:

$$S_i[j] = P(X_j = T \mid X_i = T) = \frac{N_{ij}}{N_i} \quad (3)$$

where N_{ij} is the number of instances in TS_{ET} where $X_i = T$ and $X_j = T$ occur conjointly and N_i is the number of instances in TS_{ET} where $X_i = T$.

3.3 Global Learning Approach

The global approach can be summarized as follows:

Algorithm 2. Learning bow ties

Data: $TS_{FT}; TS_{ET}; TE$

Result: BT

begin

learning structure

$T_{FT} \leftarrow \text{Learning_undirected_tree_structure}(TS_{FT}, TE);$

$T_{ET} \leftarrow \text{Learning_undirected_tree_structure}(TS_{ET}, TE);$

Orient arcs in T_{FT} towards TE ;

Orient arcs in T_{ET} backwards TE ;

$T \leftarrow \{T_{FT}, T_{ET}\};$

learning parameters

$\forall X_i \in T_{FT}$ compute $P(X_i = k \mid Pa(X_i) = j)$ using equation (2)

$\forall X_i \in T_{ET}$ compute S_i using equation (3)

end

Once the bow tie is constructed (structure and parameters), we can use it to propose appropriate protective and preventive barriers. This process can also be improved by taking into account learned conditional probability tables and different severity degrees via inference mechanism [10]. In fact, we can easily, imagine a dynamic way to propose such barriers while taking into consideration available resources.

4 Illustrative Example

This section illustrates our method via an example released in *TOTAL TUNISIA company*. Due to the lack of space we will limit our example to a unique risk

relative to a major fire and explosion on tanker truck carrying hydrocarbon (*TE*). To construct the relative bow tie we have identified six events leading to *TE* (i.e. hydrocarbon gas leak (*HGL*), and source of ignition close to road (*SI*) tank valve failure (*TVF*), exhaust failure (*EF*), and construction site close to the truck parking (*CTP*)) and nine events representing its consequences (i.e pool fire(*PF*), thermal effects (*THE*), toxic effects (*TO*), production process in stop (*PPS*), thermal damage to persons (*TDP*), damage to the other trucks (*DT*), toxic damage to persons (*TODP*), damage to environment (*DE*) and late delivery (*LD*)). The training set relative to causes TS_{FT} and consequences TS_{ET} is given in table 1 where value 1 means false and 2 true.

Table 1. Training set

| | Training set relative to causes (TS_{FT}) |
|------|-----------------------------------------------------------|
| TE | 111111111111111111222221111111111111222222222211111111111 |
| EF | 112121111111111111222111111111111121112222211211111111111 |
| CTP | 122111121111111111212221111111111111111222221111111211111 |
| TVF | 111111111111111121212221111111111111121211111111111211111 |
| HGL | 1221111211111111222222111111111111121222222111111211111 |
| SI | 112111111111111112221211111111121122222212111111111111 |
| | Training set relative to consequences (TS_{ET}) |
| TE | 111111111111111122222211111111111122222222221111111111111 |
| LD | 21111111211111112222221111111111122222222121111112111111 |
| DE | 2111111111111111222222212111111111222222221221111111112 |
| TODP | 111111111111111122222212111111111222222221221111111122 |
| DT | 211111111111111122222221111111111122222221211111111122 |
| TDP | 11211111111111122222221111111211122222221211111111111 |
| PPS | 11111112111111122222211111111112222222121111121111111 |
| TO | 11111111111111112222221211111111122222221221111111111 |
| THE | 111111111111111222222211111111111222222221211111111111 |
| PF | 111111111111111122222211111111111122222221211111111111 |

4.1 Learning Bow Tie Structure

The first step to construct the structure of the bow tie (i.e algorithm 1) is to compute the mutual information between pairs of events (causes and consequences). The relative values to TS_{FT} (resp. TS_{ET}) are given in table 2 (resp. 3), those in bold represent the best configurations (e.g. the more significant causes for *TE* are *HGL* and *SI*).

Using these values, the structure learning phase of algorithm 2 generates the bow tie diagram illustrated by figures 2. This bow tie was validated by experts in Total Tunisie since it corresponds to the one they have already proposed.

4.2 Learning Bow Tie Parameters

Once the bow tie diagram is constructed, the second phase allows us to quantify it by assigning to each node in *FT* its CPT and to each node in *ET* (i.e. consequence of *TE*) its severity vector. These values are given by tables 4 and 5, respectively.

Table 2. Mutual information values relative to TS_{FT}

| | <i>TE</i> | <i>SI</i> | <i>HGL</i> | <i>TVF</i> | <i>CTP</i> | <i>EF</i> |
|------------|---------------|---------------|---------------|------------|---------------|---------------|
| <i>TE</i> | – | 0.4637 | 0.3417 | 0.0994 | 0.3115 | 0.2350 |
| <i>SI</i> | 0.4637 | – | 0.2501 | 0.0464 | 0.5758 | 0.4492 |
| <i>HGL</i> | 0.3419 | 0.2501 | – | 0.2504 | 0.1385 | 0.1082 |
| <i>TVF</i> | 0.0994 | 0.0464 | 0.2504 | – | 0.0239 | 0.0018 |
| <i>CTP</i> | 0.3115 | 0.5758 | 0.1385 | 0.0239 | – | 0.3205 |
| <i>EF</i> | 0.2350 | 0.4492 | 0.1082 | 0.0018 | 0.3205 | – |

Table 3. Mutual information values relative to TS_{ET}

| – | <i>TE</i> | <i>PF</i> | <i>THE</i> | <i>TO</i> | <i>PPS</i> | <i>TDP</i> | <i>DT</i> | <i>TODP</i> | <i>DE</i> | <i>LD</i> |
|-------------|-----------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| <i>TE</i> | – | 0.7384 | 0.5727 | 0.5210 | 0.5210 | 0.4770 | 0.4386 | 0.4386 | 0.4386 | 0.4770 |
| <i>PF</i> | 0.7384 | – | 0.6731 | 0.6206 | 0.6206 | 0.5758 | 0.5366 | 0.5366 | 0.5366 | 0.5758 |
| <i>THE</i> | 0.5210 | 0.6731 | – | 0.5937 | 0.5937 | 0.7163 | 0.6606 | 0.4949 | 0.4949 | 0.5403 |
| <i>TO</i> | 0.5210 | 0.6206 | 0.5937 | – | 0.5282 | 0.4757 | 0.4311 | 0.7344 | 0.7344 | 0.4757 |
| <i>PPS</i> | 0.5210 | 0.6206 | 0.5937 | 0.5282 | – | 0.4757 | 0.4311 | 0.4311 | 0.4311 | 0.8046 |
| <i>TDP</i> | 0.4770 | 0.5758 | 0.7163 | 0.4757 | 0.4757 | – | 0.4913 | 0.3804 | 0.3804 | 0.4241 |
| <i>DT</i> | 0.4386 | 0.5366 | 0.6606 | 0.4311 | 0.4311 | 0.4913 | – | 0.5604 | 0.5604 | 0.4913 |
| <i>TODP</i> | 0.4386 | 0.5366 | 0.4949 | 0.7344 | 0.4311 | 0.3804 | 0.5604 | – | 0.7140 | 0.3804 |
| <i>DE</i> | 0.4386 | 0.5366 | 0.4949 | 0.7344 | 0.4311 | 0.3804 | 0.5604 | 0.7140 | – | 0.4913 |
| <i>LD</i> | 0.4770 | 0.5758 | 0.5403 | 0.4757 | 0.8046 | 0.4241 | 0.4913 | 0.3804 | 0.4913 | – |

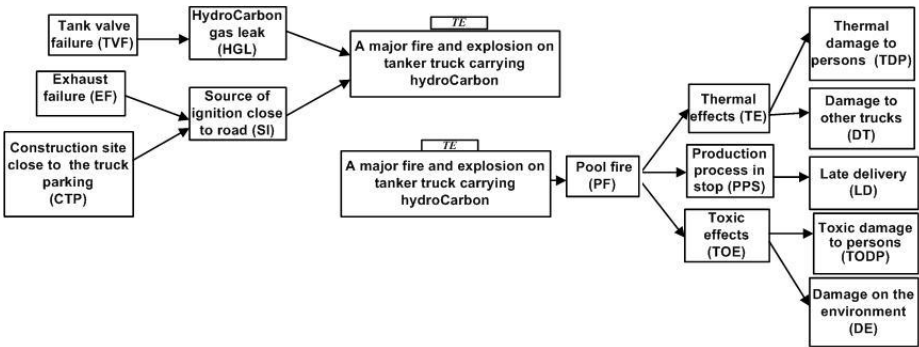


Fig. 2. Resulted bow tie diagram

Regarding the preventive and protective barriers, experts can simply use the generated graphical and numerical components in order to implement them. Note that the numerical component can be useful in this task since it will inform experts about the strength of different links in the bow tie.

Table 4. Numerical component relative to *FT*

| | | | | |
|--------------------------|-----------|----------------------|----------------------|---------------------------------|
| a, b | SI, HGL | \overline{SI}, HGL | SI, \overline{HGL} | $\overline{SI}, \overline{HGL}$ |
| $\hat{P}(TE = T a, b)$ | 0.8462 | 0.375 | 0.6 | 0.0294 |
| a, b | EF, CTP | \overline{EF}, CTP | EF, \overline{CTP} | $\overline{EF}, \overline{CTP}$ |
| $\hat{P}(SI = T a, b)$ | 0.875 | 0.333 | 0.75 | 0.0571 |
| a | TVF | | \overline{TVF} | |
| $\hat{P}(HGL = T a)$ | 0.766 | | 0.111 | |
| $\hat{P}(EF)$ | 0.7593 | | | |
| $\hat{P}(CTP)$ | 0.7407 | | | |
| $\hat{P}(TVF)$ | 0.8519 | | | |

Table 5. Severity degrees relative to *ET*

| | | | | | |
|-------------|-----------|-----------|------------|-----------|------------|
| - | <i>TE</i> | <i>PF</i> | <i>THE</i> | <i>TO</i> | <i>PPS</i> |
| <i>TE</i> | - | - | - | - | - |
| <i>PF</i> | 0.8824 | - | - | - | - |
| <i>THE</i> | - | 0.9375 | - | - | - |
| <i>TO</i> | - | 0.9375 | - | - | - |
| <i>PPS</i> | - | 0.9375 | - | - | - |
| <i>TDP</i> | - | - | 0.9444 | - | - |
| <i>DT</i> | - | - | 0.9444 | - | - |
| <i>TODP</i> | - | - | - | 0.9474 | - |
| <i>DE</i> | - | - | - | 0.9474 | - |
| <i>LD</i> | - | - | - | - | 0.9474 |

5 Conclusion

This paper proposes a new approach to construct bow tie diagrams which reflect the real behavior of exiting system i.e which are not exclusively based on the expert knowledge. Our approach is divided into two parts, first a learning algorithm is proposed to construct the whole scenario from *IE* to *ME*, and the second is a numerical component allowing us to characterize the impact of different causes on the top event *TE* and to study its repercussion while considering its severity and those of its consequences. To learn our bow tie diagrams we have proposed the algorithm [2]. This latter uses Chow and Liu [3] algorithm, this choice was motivated by the fact that this algorithm provide us a spanning tree from a training set, which characterizes both FT and ET structure learning.

This approach, lies in a recent work that we have recently proposed in order to implement a new process-based approach relative to an integrated management system: Quality, security, environment (QSE) [1]. More precisely, the bow tie diagrams are considered as input to define the appropriate management plan QSE [2]. As future work we propose to overcome the problem related to the

implementation of preventive and protective barriers. In fact, the choice of the appropriate barriers is not an easy task, since it depends on many criteria such as effectiveness, reliability, availability and cost [5]. Thus their definition from experts experience without any consideration of real data, as done in actual applications, may affect their quality since it seems unrealistic to suggest static recommendations in real dynamic systems. Thus our idea is to benefit from the numerical component, that we have added to bow ties, in order to enable experts to interact with the system in a real time via inference algorithms [10] and multicriteria analysis.

References

1. Badreddine, A., Ben Romdhane, T., Ben Amor, N.: A new process based approach for implementing an integrated management system: Quality, security, environment. In: Proc. The 2009 International Conference on Industrial Engineering, IMECS, pp. 1742–1747 (2009)
2. Badreddine, A., Ben Romdhane, T., Ben Amor, N.: A Multi-objective Approach to Implement an Integrated Management System: Quality, Security, Environment. In: International Conference of Computational Intelligence and Intelligent Systems, pp. 75–81. Londres, U.K (2009)
3. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory* 14(3), 462–467 (1968)
4. Cormen, T.H., Leiserson, C.E., Rivest, R.R.: *Introduction to Algorithms*. MIT Press, Cambridge (1990)
5. Couronneau, J.C., Tripathi, A.: Implementation of the new approach of risk analysis in france. In: 41st International Petroleum Conference, Bratislava (2003)
6. Delvosallea, C., Fieveza, C., Piparta, A., Casal, F.J., Planasb, E., Christouc, M., Mushtaqc, F.: Identification of reference accident scenarios in SEVESO establishments. *Reliability Engineering and System Safety* 90, 238–246 (2005)
7. Kurowicka, D., Cooke, R., Goossens, L., Ale, B.: Expert judgment study for placement ladder bowtie. *Safety Science* 46, 921–934 (2008)
8. Marina, M., Michael, I.: Learning with Mixtures of Trees. *Journal of Machine Learning Research* 1, 1–48 (2000)
9. Markowski, S., Sam Mannan, M., Bigoszevska, A.: Fuzzy logic for process safety analysis. *Journal of Loss Prevention in the Process Industries* 22(6), 1–8 (2009)
10. Pearl, J.: Fusion propagation and structuring in belief networks. *Artificial Intelligence* 29, 241–288 (1986)

Feature Selection and Occupancy Classification Using Seismic Sensors

Arun Subramanian¹, Kishan G. Mehrotra¹, Chilukuri K. Mohan¹,
Pramod K. Varshney¹, and Thyagaraju Damarla²

¹ Department of Electrical Engineering and Computer Science, Syracuse University,
Syracuse, NY 13244, USA

² Army Research Laboratory, Adelphi, MD 20783, USA

Abstract. In this paper^[1], we consider the problem of indoor surveillance and propose a feature selection scheme for occupancy classification in an indoor environment. The classifier aims to determine whether there is exactly one occupant or more than one occupant. Data are obtained from six seismic sensors (geophones) that are deployed in a typical building hallway. Four proposed features exploit amplitude and temporal characteristics of the seismic time series. A neural network classifier achieves performance ranging between 77% to 95% on the test data, depending on the type of construction of the location in the building being monitored.

1 Introduction

Automatic surveillance is essential for many scenarios that require remote monitoring. Several sensors of different modalities (e.g., video, acoustic, infra-red and seismic) may be deployed to monitor areas such as international borders or cleared buildings where a sustained presence of personnel is not feasible. In some cases, constraints on logistics do not permit the use of information rich sensors such as video. We consider one such scenario for indoor surveillance and investigate the use of seismic sensors for occupancy classification.

Detecting the presence of objects using seismic sensors can be categorized into two broad areas: detection of (i) humans, and (ii) other objects (such as vehicles). Detection of vehicles is easier to accomplish than detection of humans, since the seismic signal from a vehicle has a higher signal to noise ratio (see [1,2,3]). The harmonic signatures of two vehicles of the same type are consistent, and those of different vehicle types are generally distinguishable. In contrast, classification of human occupancy using footstep signals is difficult in that different people walk with different gaits and at varying pace, resulting in vibration patterns that are difficult to identify uniquely.

To detect occupants based on footstep signal processing using seismic sensors, researchers use approaches such as auto-regressive modeling, signal moments,

¹ This research was sponsored by Army Research Laboratory and was accomplished under Cooperative Agreement No. W911NF-07-2-0007. It was also supported in part by ARO grant W911NF-09-1-0244.

time-scale analysis and explicit experimental modeling [4,5,6,7]. Using a copula based approach, Iyengar et al. [8] fuse signals from acoustic and seismic sensors for footstep detection.

Although the detection of the presence of humans has been successfully addressed by such researchers, finer-grain analysis is much more difficult. One such challenging task, addressed in this paper, is the determination of whether one or more people are walking together in the environment being monitored. This is because people walking together have a psychological tendency to walk in “lock step” [9]. Since the raw signals are not easily classifiable, an important step is to derive suitable features that can assist classification. One of the main contributions of this work is to develop features that enable us to distinguish between the presence of one or more than one person in the given region of interest.

In Section 2, we describe the data collection process. The raw data is preprocessed to extract useful information, and to eliminate noise, using a methodology based on empirical mode decomposition (EMD), described in Section 3. Our approach for feature selection is presented in Section 4. Section 5 contains the classification results and Section 6 contains our conclusions.

2 Data Collection

Six GS 20DX geophones were used for the purpose of data collection. The geophones are designed to be floor mounted. The sensors were configured as a linear array. Data was collected in two (different) building hallways of similar construction. The sensors were placed along the long edge of the hallway. The distance between two adjacent sensors was maintained at 5ft. Data was acquired using a 16 bit A/D converter at a sampling rate of 5kHz. The raw signal was uniformly down-sampled to 1024 per second. The approximate duration of the data collected per trial is 12 seconds.

Multiple persons participated in the data collection. The footstep data thus collected consists of 120 single-person trials (i.e., a given trial has exactly one participant walking along the hallway) and 120 two-person trials (a given trial has exactly two participants walking along the hallway). Each dataset consists of 60 trials from Building 1 and 60 trials from Building 2.

3 Overview of the Preprocessing

In order to derive reliable features from pseudo-periodic signals like footsteps, it is of interest to extract a “clean” envelope corresponding to footfalls. We have used empirical mode decomposition (EMD) [10] for this purpose. EMD is a data-driven decomposition technique that captures oscillations at several scales. Each of these scales is called a *mode* and the function corresponding to a mode is called an intrinsic mode function (IMF). In order to capture interesting characteristics of footsteps and eliminate inherent noise present in the data, we have used only a small subset of these modes based on the total variation (TV) norm [6] such that the sum of TV norm of the selected modes is 90% of the total

TV norm. For a time-series of length T , these modes are treated as random vectors with T observations. These vectors are linearly combined, using the first principal component². The raw data is denoted by $y_{i,j}^{(k)}(t)$ and the processed data so obtained is denoted by $x_{i,j}^{(k)}(t)$ for $i = 1, \dots, 6, j = 1, \dots, 120, k = 1, 2, t = 1, \dots, T$; where i represents the sensor, j refers to trial, k refers to number of persons, and t is the time index. This processed data is used in the rest of the paper to extract features for classification.

The effect of this preprocessing is seen in Fig. 3. The EMD based processing extracts a faithful envelope of the raw signal revealing the salient features of each footfall.

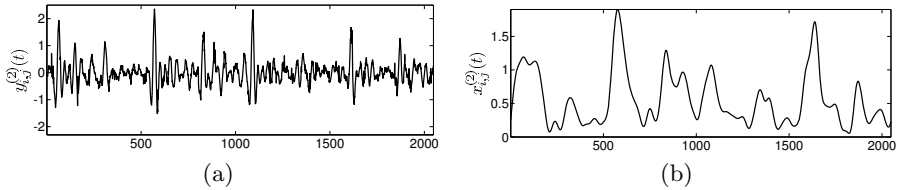


Fig. 1. A comparison of a two-second segment of the raw signal $y_{i,j}^{(2)}(t)$ and the result of envelope extraction, $x_{i,j}^{(2)}(t)$, for the 2 persons case

4 Feature Selection

In this section, we present four features that were employed to achieve classification between presence of one person versus two persons. We assume that the detection has already been accomplished successfully, i.e., *it is certain that at least one occupant is present*.

The seismic signal of a sensor decays rather rapidly as person(s) move away from the sensor. For each sensor separately, we, therefore, extract a five second segment of the “useful” data as proposed below; the discarded data is deemed to be non-informative. For each i, j, k , first we find the time index where $x_{i,j}^{(k)}$ is maximum and extract data for all time indices that are contained in $\pm t_o$ on either side of the index where the maximum occurs. We choose $t_o = 2.5 \times 1024$, because, as stated earlier, we collect 1024 samples per second. For instance, if t_{max} was the time when $x_{i,j}^{(k)}$ takes it’s maximum value, then the extracted data will be

$$\{x_{i,j}^{(k)}(t) : t = t_{max} - t_o, t_{max} - t_o + 1, \dots, t_{max} + t_o\}.$$

However, for notational convenience we renumber the time indices and denote the extracted data as

$$\{z_{i,j}^{(k)}(t) : t = 0, 1, \dots, 2t_o\},$$

² We have observed that about 80% of the signal energy is contained in the first principal component.

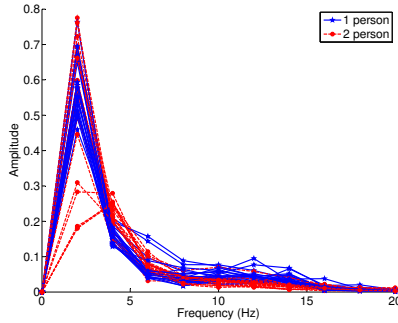


Fig. 2. Periodogram comparison of 1 person and 2 person trials

for all values of (i, j, k) . Note that this data will be extracted from the early part of $\{x_{i,j}^{(k)}(t)\}$ for the first sensor ($i = 1$) and later part for the sixth sensor ($i = 6$). In the following subsections, features are extracted from these ‘extracted’ data sets only.

4.1 Periodogram and Autocorrelation

Footsteps are quasi-periodic signals. Typically, the periodogram and autocorrelation function (ACF) are used in determining signal periodicity [2, 3, 11]. But, analysis by Houston and McGaffigan [9] shows that the use of spectral measures is not useful for counting the number of personnel in the region of interest. In the following discussion, we note that while the periodogram and autocorrelation are not directly useful for our occupancy classification problem, nevertheless they provide some valuable insights. Fig. 2 shows a comparison of the periodograms of 20 trials of the one person and two person cases. The figure suggests that the periodogram *alone* may be insufficient for the classification task at hand.

We expect that “heel-toe” transitions will be borne out in the single occupant case and will be blurred in the case of two or more occupants. In order to find support for this argument, we plotted the mean of the ACF across sensors for a single period (Fig. 4.1). We use the peak frequency from the periodogram to obtain the period of the ACF.

Clearly, the ACF for one person goes through a typical pattern of local maxima and minima as heel and toe spikes align with each other. When the ACF for the first period are averaged over all the training data for all sensors in the one-person case, we obtain the ACF *template* seen in Fig. 3(a). Therefore, the feature extraction procedure is,

1. Form the template (Fig. 3(a))
2. Calculate the area which measures the difference between the template and the ACF for a given trial (see Fig. 4). This difference is the mean-square error between the two curves and is denoted as $MSE_{i,j}^{(k)}$, which is the first of the four features that we use for occupancy classification.

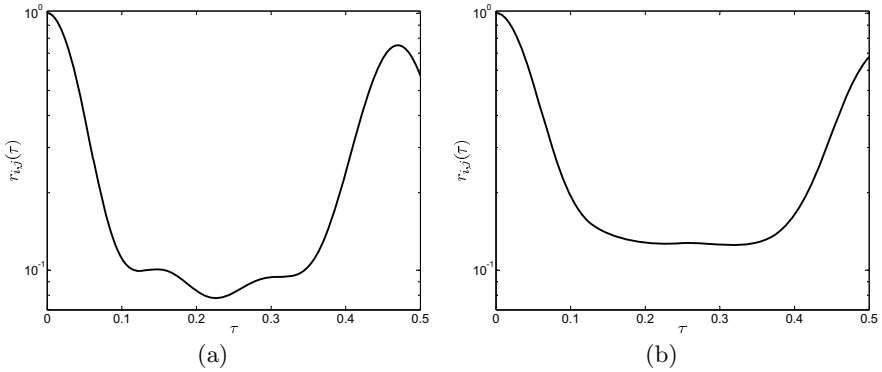


Fig. 3. The autocorrelation function for a single period averaged over all sensors. (a) 1 person. (b) 2 persons. X-axis τ is in seconds. Y-axis is $r_{i,j}(\tau)$.

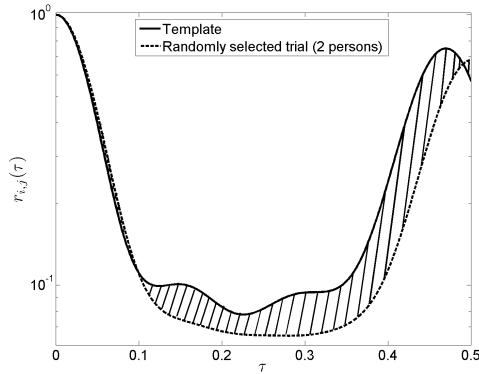


Fig. 4. Comparison of autocorrelation functions. The shaded region shows the difference in area which, when integrated, gives the mean-square error between the template and the ACF for the given trial. X-axis: τ , Y-axis: $r_{i,j}(\tau)$.

4.2 Signal Energy

Variance is a measure of signal energy and serves as an intuitive indicator of occupancy. That is, when the number of occupants is large, the signal energy will be large. Using the ‘extracted’ data described earlier we calculate the variance as described below.

$$[S_{i,j}^{(k)}]^2 = \frac{1}{2t_o} \sum_{t=0}^{2t_o} (z_{i,j}^{(k)}(t) - \overline{z_{i,j}^{(k)}})^2, \tag{1}$$

where $\overline{z_{i,j}^{(k)}}$ is the sample mean (of the $2t_o$ observations).

These variances are averaged by taking their mean across all six sensors, and

$$\mathfrak{S}_j^{(k)} = \frac{1}{6} \sum_{i=1}^N \left[S_{i,j}^{(k)} \right]^2 \quad (2)$$

is used as one of the features to be used for classification.

4.3 Ratio of Time Spent in States

We expect that footsteps produce a quasi-periodic time series when one person walks whereas footsteps for two persons will be irregular and spread out. For two or more persons this “footstep state” will occupy a greater proportion of a window of fixed duration. Based on this criterion, the feature $R(\text{states})$ is calculated for each value of (i, j, k) as described below. For notational convenience we do not use j and k below.

As stated earlier, we use the ‘extracted’ data only. We move a sliding window (one-tenth of a second long) over this data, calculate the average of the x -values within the window, and compare it with a threshold η to calculate

$$d_{pi} = \begin{cases} 1 & \text{if the average of the observations within the window is } > \eta \\ 0 & \text{if the average of the observations within the window is } \leq \eta \end{cases} \quad (3)$$

where $p = 1, \dots, P$ and P represents the number of times the sliding window fits over the data. In the ideal situation, each ‘1’ represents the “Footstep state” and each ‘0’ represents the “Silence state.” The ratio

$$R(\text{states}) = \frac{\sum_{p,i} d_{pi}}{\sum_{p,i} (1 - d_{pi})} \quad (4)$$

captures the ratio of time spent in these states versus not in the state. These ratios, the third feature, are obtained for all values of (j, k) and are denoted as $R(\text{states})_j^{(k)}$.

Threshold η is calculated from the first 0.25 seconds of the x data which represents the background process. Let μ_b and σ_b denote the sample mean and standard deviation calculated from this initial 0.25 second period. The threshold is then set as,

$$\eta = \mu_b + 3\sigma_b \quad (5)$$

4.4 Cross-Correlation and Wavelet Based Feature

The cross-correlation sequence between adjacent sensors is one way to measure the similarity between the data collected by two sensors. We considered the cross-correlation between the two nearest neighbors of a given sensor; for example, two nearest neighbors of sensor 2 are sensor 1 and sensor 3, respectively. As before the “extracted” data is obtained for sensors $i = 2, 3, 4, 5$ and the following correlation functions are computed:

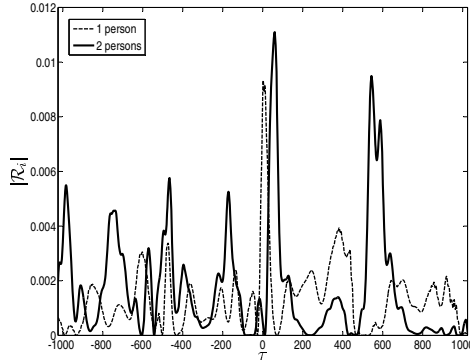


Fig. 5. Typical plot of $|\mathcal{R}_i|$ vs. τ

$$r_{i+}(\tau) = \text{Corr}(z_{i,j}^{(k)}(t), z_{(i+1),j}^{(k)}(t + \tau)) \tag{6}$$

$$r_{i-}(\tau) = \text{Corr}(z_{i,j}^{(k)}(t), z_{(i-1),j}^{(k)}(t + \tau)) \tag{7}$$

$$r_{i\pm}(\tau) = \text{Corr}(z_{(i+1),j}^{(k)}(t), z_{(i-1),j}^{(k)}(t + \tau)) \tag{8}$$

$$\mathcal{R}_i(\tau) = \sqrt{r_{i+}(\tau)r_{i-}(\tau)} - r_{i\pm}(\tau) \tag{9}$$

where τ extends from -1024 to 1024 (i.e., up to a shift of 1 second in both directions).

We expect that, in the *ideal* situation, $\mathcal{R}_i = 0$ for the 1-person case and $\mathcal{R}_i \neq 0$ when two or more persons are walking. In practice, \mathcal{R}_i is expected to take a small value for the 1-person case and a large value in the 2 or more persons case. In Fig. 5, we observe wide major peaks and thinner auxiliary peaks in the two-person case. In other words, the 2-person information is contained at different scales. This suggests that wavelet decomposition would be useful in extracting useful information from $\mathcal{R}_i(\tau)$.

We chose the Mexican hat wavelet and calculated the continuous wavelet transform (scalogram) of $\{\mathcal{R}_i(\tau) : \tau = -1024 \text{ to } 1024\}$,

$$C_i(a, b) = \int_{\text{all}\tau} \mathcal{R}_i(\tau)\psi(\tau - b, a)d\tau, \tag{10}$$

$$\psi(\tau, a) = \frac{1}{\sqrt{2\pi}a^3} \left(1 - \frac{\tau^2}{a^2}\right) \exp\left(-\frac{\tau^2}{2a^2}\right)$$

We note that, in theory, the scale a can have an infinite number of values. For numerical computation, we chose the maximum value of a to be $1/(2f^*)$ where f^* is the peak frequency from the periodogram of the extracted data.

Finally, a summary statistics can be measured in terms of the average S_C ,

$$\overline{C}(a, b) = \frac{1}{4} \sum_{i=2}^5 |C_i(a, b)| \quad (11)$$

$$S_C = \sum_b \max_a \overline{C}(a, b) \quad (12)$$

or use this feature separately for each sensor as:

$$S_{C_i} = \sum_b \max_a |C_i(a, b)|. \quad (13)$$

5 Classification Procedure and Results

Section 4 discussed the features investigated for classification problem of determining one vs. two occupants in a building hallway. There are two possible fusion schemes for the selected features:

1. Use the 4-dimensional feature vector $F = [MSE_j^{(k)}, \mathfrak{S}_j^{(k)}, R(\text{states})_j^{(k)}, S_C]$ where the individual features have been “fused” across sensors by taking the mean
2. Use a 17-dimensional vector comprising of $MSE_{i,j}^{(k)}$, $[S_{i,j}^{(k)}]^2$, $R(\text{states})_j^{(k)}$, and S_{C_i} for each i . Note that there is only one value for the third feature and 4 values for the fourth feature giving a total of $(6 + 6 + 1 + 4 = 17)$ -dimensional vector. In this case the fusion is achieved by dimensionality reduction using principal component analysis (PCA) retaining those combined features that capture 98% of the total variation. PCA is performed prior to classification.

In both cases, a neural network classifier is used with one hidden layer of 6 nodes³. Classification performance is analyzed for the following five cases,

Case 1 Train on Building 1 data, test on Building 2 data

Case 2 Train on Building 1 data, test on Building 1 data⁴.

Case 3 Train on Building 2 data, test on Building 2 data⁴.

Case 4 Train on Building 2 data, test on Building 1 data.

Case 5 Mixed, i.e., combine data from both buildings by random permutation and use half the dataset to train and remaining half to test

Tables 1 and 2 contain the results on test data; cases 2, 3 and 5 are the average of 10 iterations. We conclude that, in general, the classification performance is very good. Best performance is observed for Case 3 (Building 2). This indicates that selected features are good. If, somehow, we can improve the method of sensor data collection, then the classification performance will further improve.

³ We experimented with 3 to 9 nodes in the hidden layer. Best training performance was observed with 6 nodes.

⁴ Training is done using a randomly selected set of 90 trials out of the 120.

Table 1. Classification results: 4-dimensional feature vector

| True/Error classes | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
|-----------------------|--------|--------|--------|--------|--------|
| C_1 true (TP) | 119 | 29 | 29 | 78 | 114 |
| C_2 true (TN) | 91 | 24 | 28 | 108 | 103 |
| Type I error (FA) | 1 | 1 | 1 | 42 | 6 |
| Type II error (M) | 29 | 6 | 2 | 12 | 17 |
| Performance (P) | 87.50% | 88.33% | 95.00% | 77.50% | 90.41% |

Table 2. Classification results: 17-dimensional feature vector followed by PCA

| True/Error classes | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
|-----------------------|--------|--------|--------|--------|--------|
| C_1 true (TP) | 120 | 30 | 30 | 90 | 114 |
| C_2 true (TN) | 91 | 23 | 28 | 99 | 108 |
| Type I error (FA) | 0 | 0 | 0 | 30 | 6 |
| Type II error (M) | 29 | 7 | 2 | 21 | 12 |
| Performance (P) | 87.91% | 88.33% | 96.67% | 78.75% | 92.50% |

TP : True positives, TN : True negatives, FA : False alarms, M : Misses,
 $P = (TP + TN)/(TP + TN + FA + M) \times 100\%$

The ‘good’ classification performance of the classifiers trained on the data from one building and tested on the same building supports the above conclusion.

To assess the contribution of each feature *alone* we experimented with Case 5. It was observed that the third feature ($R(\text{states})_j^{(k)}$) contributes significantly to improving the classification performance (about 70%), whereas features $MSE_{i,j}^{(k)}$ and S_{C_i} make similar amount of contributions. The similarity in performance between these two features is not surprising because they are both derived from correlation sequences. Since the rationale for using S_{C_i} depends on the linear configuration of sensors, one may consider dropping this feature for a simpler system design under more general topology of sensor deployment. We have noticed that in spite of dropping this feature we get a classification performance of 89.13% for Case 5.

6 Concluding Remarks

We have addressed the task of distinguishing between the presence of one vs. two persons in a visually unobservable indoor region of interest, using data obtained from seismic sensors. After exploring multiple alternatives, we identified four features as being capable of assisting classification with a high degree of reliability. Classification was achieved using a neural network classifier.

Although the current suite of four features has yielded very good performance, further improvements may be possible by exploring new feature extraction approaches. Performance gains may be obtained by using non-linear dimensionality

reduction schemes [12] as opposed to principal component analysis, which is linear. Improved classification as well as finer classification is also expected using other modalities; such as the acoustic sensors. Since the data are vibrational in nature, the signal processing and feature extraction algorithms developed in this paper can also be applied to data collected using accelerometers and acoustic sensors.

References

1. Dibazar, A.A., Park, H.O., Berger, T.W.: The application of dynamic synapse neural networks on footstep and vehicle recognition. In: Proc. International Joint Conference on Neural Networks IJCNN 2007, August 12–17, pp. 1842–1846 (2007)
2. Li, D., Wong, K.D., Hu, Y.H., Sayeed, A.M.: Detection, classification, and tracking of targets. *IEEE Signal Processing Magazine* 19(2), 17–29 (2002)
3. Tian, Y., Qi, H., Wang, X.: Target detection and classification using seismic signal processing in unattended ground sensor systems. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002), vol. 4, p. 4172 (2002)
4. Bland, R.E.: Acoustic and seismic signal processing for footstep detection. Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science (2006)
5. Succi, G., Clapp, D., Gampert, R., Prado, G.: Footstep detection and tracking. In: Proc. SPIE - Int. Soc. Opt. Eng (USA), USA, vol. 4393, pp. 22–29 (2001)
6. Subramanian, A., Iyengar, S.G., Mehrotra, K.G., Mohan, C.K., Varshney, P.K., Damarla, T.: A data-driven personnel detection scheme for indoor surveillance using seismic sensors. In: Carapezza, E.M. (ed.) Unattended Ground, Sea, and Air Sensor Technologies and Applications XI. SPIE, vol. 7333, p. 733315 (2009)
7. Sabatier, J.M., Ekimov, A.E.: A review of human signatures in urban environments using seismic and acoustic methods. In: Proc. IEEE Conference on Technologies for Homeland Security, May 12–13, pp. 215–220 (2008)
8. Iyengar, S.G., Varshney, P.K., Damarla, T.: On the detection of footsteps based on acoustic and seismic sensing. In: Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers ACSSC 2007, November 4–7, pp. 2248–2252 (2007)
9. Houston, K.M., McGaffigan, D.P.: Spectrum analysis techniques for personnel detection using seismic sensors. In: Carapezza, E.M. (ed.) Unattended Ground Sensor Technologies and Applications V. SPIE, vol. 5090, pp. 162–173 (2003)
10. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. In: Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 454(1971), pp. 903–995 (1998)
11. Vlachos, M., Yu, P.S., Castelli, V., Meek, C.: Structural periodic measures for time-series data. *Data Mining and Knowledge Discovery* 12(1), 1–28 (2006)
12. Lee, J.A., Verleysen, M.: *Nonlinear Dimensionality Reduction*. Springer, Heidelberg (2007)

Extending Metric Multidimensional Scaling with Bregman Divergences

Jigang Sun, Malcolm Crowe, and Colin Fyfe

University of the West of Scotland

{Jigang.Sun,Malcolm.Crowe,Colin.Fyfe}@uws.ac.uk

Abstract. We investigate multidimensional scaling with Bregman divergences and show that the Sammon mapping can be thought of as a truncated Bregman multidimensional scaling (BMDS). We show that the full BMDS improves upon the Sammon mapping on some standard data sets and investigate the reasons underlying this improvement. We then introduce two families of BMDS which use opposite strategies to create good mappings of standard data sets and investigate these opposite strategies analytically.

1 Introduction

The quantity and dimensionality of data often captured automatically has been growing exponentially over the last decades. We are now in a world in which the extraction of information from data is of paramount importance. One means of doing this is to project the data onto a low dimensional manifold and allow a human investigator to search for patterns in this projection by eye. This is obviously based on the assumption that the low dimensional projection can capture relevant features of the high dimensional data set; the earliest methods for this type of analysis were principal component analysis and factor analysis, the former giving the linear projection which is closest to the original data while the latter tries to capture specific relevant features of the data set in which “relevant” often has to accord with some human intuition. However such projections are linear and there is no *a priori* reason that a high dimensional data set should lie on a linear subspace.

Fortunately however high dimensional data sets often lie on a (non-linear) embedded manifold of the data set where a manifold generally is thought of as a topological space which is locally Euclidean. Finding such embedded manifolds is generally not a simple closed-form process however, unlike finding linear subspaces. The alternatives are to find the manifold and then project the data onto this manifold or to directly attempt to ascertain the projections of the data.

This paper deals with one group of methods for finding the projections directly, multidimensional scaling (MDS). In the remainder of this section, metric multidimensional scaling (MMDS) and Bregman divergences will be briefly reviewed. In section 2, MMDS is generalised to BMDS: as an example, the classical Sammon mapping is extended to a Bregmanised version, and a comparative

study is given, following which a criterion for base convex functions for Bregman divergences is proposed. In section 3, two groups of Bregman divergences are created and compared with each other. Finally section 4 concludes with future work being suggested.

1.1 Metric Multidimensional Scaling

We will call the space into which the data is projected the latent space: multidimensional scaling creates one latent point for every data sample. MMDS is a group of methods that keep interpoint Euclidean distances in latent space as close as possible to the original distances in data space, which is good at maintaining the global structure of the manifold. If we use D_{ij} to represent the distance between points X_i and X_j in data space, L_{ij} to represent the distance in latent space between the corresponding mapped points Y_i and Y_j , and the size of data set is N , the stress functions of MMDS can be generalised as

$$E_{MMDS}(Y) = \frac{1}{C} \sum_{i=1}^N \sum_{j=i+1}^N (L_{ij} - D_{ij})^2 W(D_{ij}) \tag{1}$$

where C is a normalisation scalar which does not affect the embedding, and $W(D_{ij}) \geq 0$ is a monotonically decreasing function of the distances in data space. This is sometimes used to emphasise local distances.

Example 1. If we set $C = 1$ and $W(D_{ij}) = 1$, eq(1) becomes linear multidimensional scaling (LMMDS)

$$E_{LMMDS}(Y) = \sum_{i=1}^N \sum_{j=i+1}^N (L_{ij} - D_{ij})^2 \tag{2}$$

Example 2. If we set $C = \sum_{i=1}^N \sum_{j=i+1}^N D_{ij}$ and $W(D_{ij}) = D_{ij}^{-1}$, eq(1) becomes the nonlinear Sammon mapping [4]

$$E_{Sammon}(Y) = \frac{1}{\sum_{i=1}^N \sum_{j=i+1}^N D_{ij}} \sum_{i=1}^N \sum_{j=i+1}^N \frac{(L_{ij} - D_{ij})^2}{D_{ij}} \tag{3}$$

1.2 Bregman Divergence

Consider a strictly convex function $F : S \rightarrow \Re$ defined on a convex set $S \subset \Re^d$. A Bregman divergence [1] between two points, \mathbf{p} and $\mathbf{q} \in S$, is defined to be

$$d_F(\mathbf{p}, \mathbf{q}) = F(\mathbf{p}) - F(\mathbf{q}) - \langle (\mathbf{p} - \mathbf{q}), \nabla F(\mathbf{q}) \rangle \tag{4}$$

where the angled brackets indicate an inner product and $\nabla F(\mathbf{q})$ is the derivative of F evaluated at \mathbf{q} . This can be viewed as the difference between $F(\mathbf{p})$ and its truncated Taylor series expansion around \mathbf{q} . Thus it can be used to ‘measure’ the

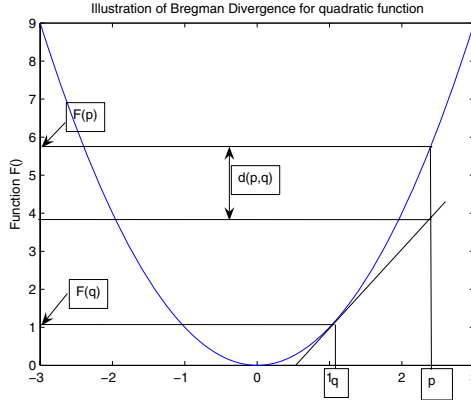


Fig. 1. The divergence is the difference between $F(p)$ and the value of $F(q) + (p - q)\nabla F(q)$

convexity of F : Figure 1 illustrates how the Bregman divergence is the difference between $F(\mathbf{p})$ and the value which would be reached at \mathbf{q} with a linear estimate, $\nabla F(\mathbf{q})$, for the curvature at \mathbf{q} .

Example 1. The squared Euclidean distance is a special case of the Bregman divergence in which $F(\cdot) = \|\cdot\|^2$, the squared Euclidean norm

$$\begin{aligned} d_F(\mathbf{p}, \mathbf{q}) &= \|\mathbf{p}\|^2 - \|\mathbf{q}\|^2 - \langle \mathbf{p} - \mathbf{q}, \nabla F(\mathbf{q}) \rangle \\ &= \|\mathbf{p}\|^2 - \|\mathbf{q}\|^2 - \langle \mathbf{p} - \mathbf{q}, 2\mathbf{q} \rangle \\ &= \|\mathbf{p} - \mathbf{q}\|^2 \end{aligned}$$

Example 2. The Itakura-Saito divergence [1] based on convex function $F(x) = -\log(x), x > 0$

$$IS(p, q) = \frac{p}{q} - \log \frac{p}{q} - 1 \tag{5}$$

will be used below.

Properties of Bregman divergences:

non-negativity : in (4) $d_F(\mathbf{p}, \mathbf{q}) \geq 0$ and $d_F(\mathbf{p}, \mathbf{q}) = 0 \iff \mathbf{p} = \mathbf{q}$.

non-symmetry : $d_F(\mathbf{p}, \mathbf{q}) \neq d_F(\mathbf{q}, \mathbf{p})$ except in special cases such as $F(\cdot) = \|\cdot\|^2$, the Euclidean norm.

2 Linking Metric MDS and Bregman Divergences

We could consider using Bregman divergences in either data space or latent space, however in this paper, we investigate using Bregman divergences between

L_{ij} and D_{ij} instead of the squared distance between them which we used above. We will call this a Bregmanised Metric MDS (BMMDS) whose stress function is a sum of divergences:

$$\begin{aligned}
 E_{BMMDS}(Y) &= \sum_{i=1}^N \sum_{j=i+1}^N d_F(L_{ij}, D_{ij}) \\
 &= \sum_{i=1}^N \sum_{j=i+1}^N (F(L_{ij}) - F(D_{ij}) - (L_{ij} - D_{ij})\nabla F(D_{ij})) \quad (6)
 \end{aligned}$$

We may alternatively express this as the tail of the Taylor series expansion and so we can consider

$$\begin{aligned}
 &E_{BMMDS}(Y) \\
 &= \sum_{i=1}^N \sum_{j=i+1}^N \frac{d^2 F(D_{ij})}{dD_{ij}^2} \frac{(L_{ij} - D_{ij})^2}{2!} + \frac{d^3 F(D_{ij})}{dD_{ij}^3} \frac{(L_{ij} - D_{ij})^3}{3!} + \frac{d^4 F(D_{ij})}{dD_{ij}^4} \frac{(L_{ij} - D_{ij})^4}{4!} + \dots \quad (7)
 \end{aligned}$$

We now show that linear MMDS is a Bregman divergence and that the Sammon Mapping can be seen as special case of approximation to Bregman divergences using only the first term of (7).

Linear MMDS. When $F(x) = x^2, x > 0$, eq (7) is identical to eq (2) with $C = 1$. i.e. linear MMDS can be viewed as a Bregman divergence using the underlying convex function, $F(x) = x^2$. Of course, it is well known that the solution of the linear MMDS objective is to locate the latent points at the projections of the data points onto their first principal components i.e. minimising the divergence (in this case Euclidean distance) between the data points and their projections, so what we are really doing here is to restate an accepted equivalence in terms of Bregman divergences.

Sammon Mapping. We define base convex function

$$F(x) = x \log x, x \in \mathbb{R}_{++}, \quad (8)$$

The higher order derivatives for this base function are shown in Table 1.

Then the Sammon Mapping is the first term in (7); i.e. the Sammon Mapping can be viewed as an approximation to BMMDS. However with this underlying convex function, the higher order terms do not vanish and eq(7) can be expressed as

Table 1. The higher order derivatives for this base function (8)

| $\frac{d^2 F(x)}{dx^2}$ | $\frac{d^3 F(x)}{dx^3}$ | $\frac{d^4 F(x)}{dx^4}$ | $\frac{d^5 F(x)}{dx^5}$ | $\frac{d^6 F(x)}{dx^6}$ |
|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| $\frac{1}{x} > 0$ | $-\frac{1}{x^2} < 0$ | $\frac{2!}{x^3} > 0$ | $-\frac{3!}{x^4} < 0$ | $\frac{4!}{x^5} > 0$ |

$$\begin{aligned}
 & E_{ExtSammon}(Y) \\
 = & \sum_{i=1}^N \sum_{j=i+1}^N \frac{(L_{ij} - D_{ij})^2}{2!D_{ij}} - \frac{(L_{ij} - D_{ij})^3}{3!D_{ij}^2} + \frac{2!(L_{ij} - D_{ij})^4}{4!D_{ij}^3} - \frac{3!(L_{ij} - D_{ij})^5}{5!D_{ij}^4} + \dots
 \end{aligned} \tag{9}$$

We re-express (3) to be

$$E_{Sammon}(Y) = \frac{2}{\sum_{i=1, i < j}^N D_{ij}} \sum_{i=1, i < j}^N \frac{(L_{ij} - D_{ij})^2}{2D_{ij}} = \frac{2}{\sum_{i=1, i < j}^N D_{ij}} \sum_{i=1, i < j}^N I_{ij}^{Sammon}$$

where $I_{ij}^{Sammon} = \frac{(L_{ij} - D_{ij})^2}{2!D_{ij}}$. The constant scalar is only for normalisation purposes, and does not affect the projection at all if discarded. We therefore concentrate on

$$E'_{Sammon}(Y) = \sum_{i=1, i < j}^N I_{ij}^{Sammon} \tag{10}$$

Then an ExtendedSammon mapping will take into account all of the terms

$$\begin{aligned}
 & E_{ExtSammon}(Y) \\
 = & \sum_{i=1}^N \sum_{j=i+1}^N I_{ij}^{Sammon} - \frac{(L_{ij} - D_{ij})^3}{3!D_{ij}^2} + \frac{2!(L_{ij} - D_{ij})^4}{4!D_{ij}^3} - \frac{3!(L_{ij} - D_{ij})^5}{5!D_{ij}^4} + \dots
 \end{aligned} \tag{11}$$

We will actually use the alternative and equivalent representation, (6). The question now arises as to whether, by utilising (6) and thereby implicitly incorporating the higher order terms, we gain anything.

From MMDS to BMMDs. If set $W(D_{ij}) = \frac{1}{2!} \frac{d^2 F(D_{ij})}{dD_{ij}^2}$ and $C = 1$ in eq(1), then MMDS is the first term of BMMDs (eq(7)) and BMMDs is an extension to MMDS.

2.1 The ExtendedSammon Mapping

Note that for base function eq(8) $\frac{dF(x)}{dx} = \log x + 1$. The ExtendedSammon mapping eq(9) is equivalent to

$$\begin{aligned}
 E_{ExtSammon}(Y) &= \sum_{i=1}^N \sum_{j=i+1}^N (L_{ij} \log L_{ij} - D_{ij} \log D_{ij} - (L_{ij} - D_{ij})(\log D_{ij} + 1)) \\
 &= \sum_{i=1}^N \sum_{j=i+1}^N \left(L_{ij} \log \frac{L_{ij}}{D_{ij}} - L_{ij} + D_{ij} \right)
 \end{aligned} \tag{12}$$

which is simple to implement.

It is implemented using standard gradient descent, with initialisation of the latent points' positions as the configuration found by the Sammon mapping. It is tested on the standard data set in the literature, the Swiss roll data set, shown in Figure 2(a). The rest of Figure 2 displays 2-dimensional projections by LMMDS, the Sammon and ExtendedSammon mappings. We see that linear multidimensional scaling only captures the curve - there is no ability to differentiate points which lie at the same X,Y coordinate but different Z (vertical) coordinate in Figure 2(a). The Sammon mapping does rather better while the ExtendedSammon mapping does best of all.

However this subjective comparison needs to be augmented quantitatively and with more analysis which hopefully will reveal why one projection is preferable to another. Figure 3 shows graphs giving intuitions about the projections made by LMMDS, the Sammon and ExtendedSammon mappings. In Figure 3(a) the distances D_{ij} in data space are quantised into 40 sets, $\mathcal{D}_h, h = 1, 2, \dots, 40$.

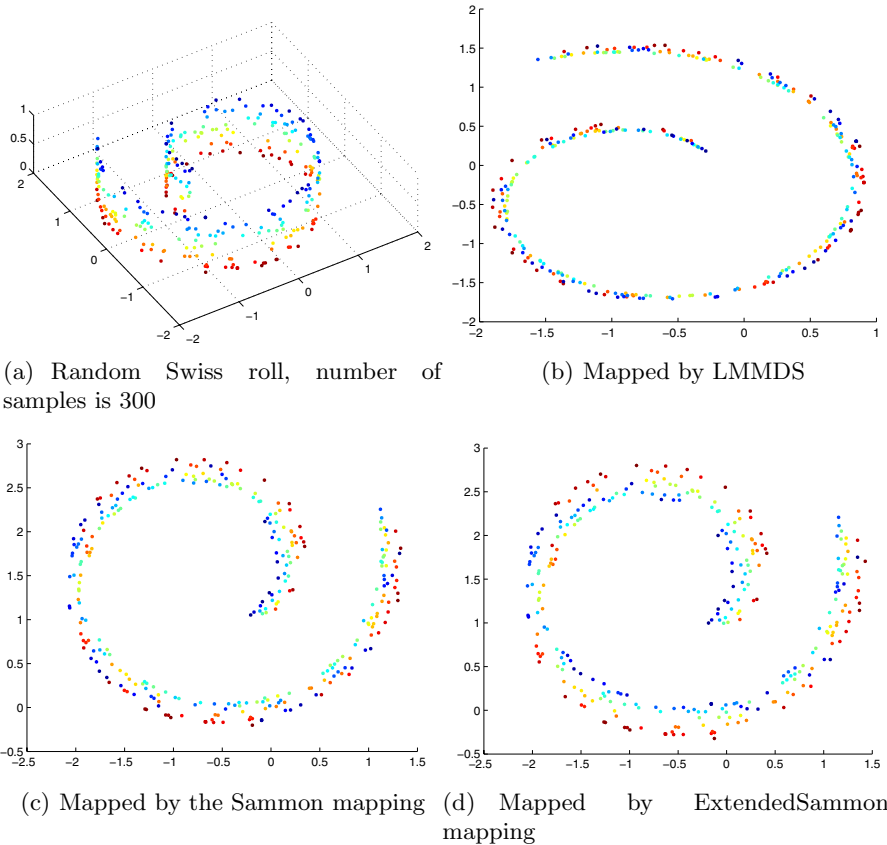


Fig. 2. Swiss roll, and projections by LMMDS, Sammon and ExtendedSammon mapping. The colours in each diagram indicate the vertical position on the original Swiss roll.

\mathcal{L}_h represents corresponding distances in latent space. $\overline{\mathcal{D}}_h$, the mean of \mathcal{D}_h , is represented on the horizontal axis. The vertical axis shows

$$\text{Mean Relative Distance} = \frac{\overline{\mathcal{L}}_h - \overline{\mathcal{D}}_h}{\overline{\mathcal{D}}_h}, \quad h = 1, 2, \dots, 40$$

where $\overline{\mathcal{L}}_h = \text{mean}(\mathcal{L}_h)$

It can be seen that distances less than 2.0 are compressed in all three mappings. Among these distances, those that are less than 1.0 are reduced most. Although it is well known that the Sammon mapping puts more stress on local distances and less stress on long distances, small distances are not projected close to their original value as expected. They are actually over compressed instead whereas long distances are preserved better. The reason for this phenomenon is that during optimisation, we can consider that short distances and long distances are competing for opportunities to go closer to their original values as much as possible. Although each individual short distance has a stronger effect than a longer one, the number of distances that are longer than 1.0 greatly surpasses the number of short ones, as shown in Figure 3(a), the histogram of the distances in the original data space. So the net effect is that the long distances are constrained as shown. That this effect is not due to an averaging of under and over compression can be seen in Figure 3(b).

Nevertheless short distances mapped by the Sammon mapping are closer to their counterpart data distances than those mapped by LMMDS, which means that local distance preservation is improved. Again the ExtendedSammon mapping makes a further improvement to the Sammon mapping although the step from Sammon to ExtendedSammon is less than from LMMDS to Sammon. While keeping local distances short, the Sammon and ExtendedSammon mapping stretch long distances slightly in turn. In Figure 3(b) we can see that the relative deviation achieved by the Sammon mapping on small distances is smaller than by LMMDS, and it is the same scenario for ExtendedSammon over Sammon, which means, on the Swiss roll data, in the sequence of LMMDS, Sammon and ExtendedSammon, controlling local distances is more and more enhanced. It also can be seen that in the same order more and more freedom is given to large distances.

2.2 Criterion for Base Convex Function Selection

Based on the above, this suggests that, for use in multidimensional scaling applications, a convex function $F(x)$ defined on \mathbb{R}_{++} should satisfy

$$\left\{ \begin{array}{l} \frac{d^2 F(x)}{dx^2} > 0, \frac{d^4 F(x)}{dx^4} > 0, \dots, \frac{d^{(2n)} F(x)}{dx^{(2n)}} > 0 \\ \frac{d^3 F(x)}{dx^3} < 0, \frac{d^5 F(x)}{dx^5} < 0, \dots, \frac{d^{(2n+1)} F(x)}{dx^{(2n+1)}} < 0, n = 1, 2, 3, \dots \end{array} \right. \quad (13)$$

Thus, if a convex function such as $F(x) = \exp(x), x > 0$ is used as the base function, even although the overall stress minimises and L_{ij} approaches D_{ij} , it does not improve local distance preservation.

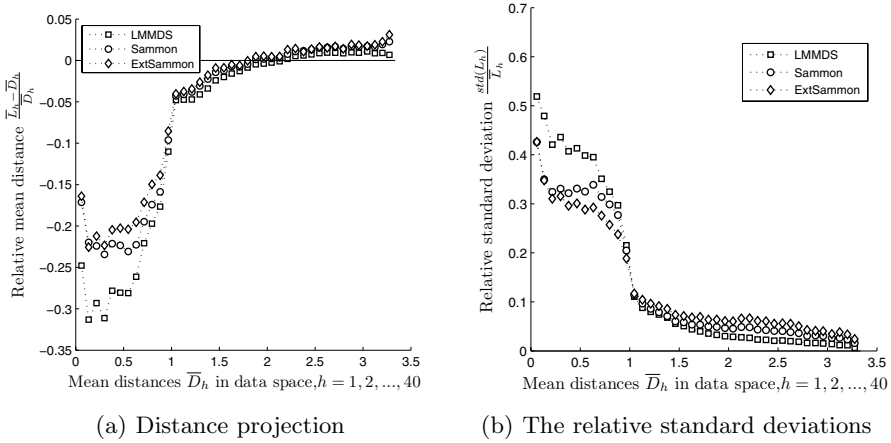


Fig. 3. Analysis of the mapping of Swiss roll by LMMDS, Sammon and ExtendedSammon

Having met (13), the second order derivative is then primarily considered to be the most important, because it is a major influence in eq(7) as shown above. It needs to be big at small distances and small at long distances. The bigger it is at local distances, the greater force concentrates on small distances. The smaller it is at long distances, the more freedom the long distances have.

3 Two Groups of Bregman Divergences

Besides eq(8) as the base convex function for the ExtendedSammon mapping, two groups of more suitable base functions that meet the criterion discussed above are proposed in Table 2. However these use different strategies for increasing focusing power: in the first group from No. 1 to No. 4 the second order derivatives are higher for small distances so focusing power is increasing; the second group only gives limited maximum values to very small distances but reduces the value at long distances. For the same long distance, the bigger λ is, the smaller the second derivative is, and thus the greater the focusing power is.

Based on the proposed base functions other BMMDS stress functions are created as follows

$F(x) = -\log(x)$: This is the Itakura-Saito (eq(5)) divergence .

$$E_{IS}(Y) = \sum_{i=1}^N \sum_{j=i+1}^N IS(L_{ij}, D_{ij}) = \sum_{i=1}^N \sum_{j=i+1}^N \left(\frac{L_{ij}}{D_{ij}} - \log \frac{L_{ij}}{D_{ij}} - 1 \right) \quad (14)$$

Table 2. Some convex functions and their derivatives, $x > 0$

| | No. | $F(x)$ | $\frac{d^2 F}{dx^2}$ | $\frac{d^3 F}{dx^3}$ | $\frac{d^n F}{dx^n}$ |
|-------------|---------------------|------------------|----------------------------|--------------------------------|--------------------------------------------------|
| FirstGroup | 1 | $x \log(x)$ | $\frac{1}{x}$ | $-\frac{1}{x^2}$ | $\frac{(-1)^n (n-2)!}{x^{n-1}}$ |
| | 2 | $-\log(x)$ | $\frac{1}{x^2}$ | $-\frac{2}{x^3}$ | $\frac{(-1)^n (n-1)!}{x^{n-1}}$ |
| | 3 | $\frac{1}{x}$ | $\frac{2}{x^3}$ | $-\frac{6}{x^4}$ | $\frac{(-1)^n (n)!}{x^{n+1}}$ |
| | 4 | $\frac{1}{x^2}$ | $\frac{6}{x^4}$ | $-\frac{24}{x^5}$ | $\frac{(-1)^n (n+1)!}{x^{n+2}}$ |
| | generic, $t \geq 1$ | $\frac{1}{x^t}$ | $\frac{t(t+1)}{x^{t+2}}$ | $-\frac{t(t+1)(t+2)}{x^{t+3}}$ | $\frac{(-1)^n t(t+1)(t+2)\dots(t+n-1)}{x^{t+n}}$ |
| SecondGroup | 5 | $e^{-\lambda x}$ | $\lambda^2 e^{-\lambda x}$ | $-\lambda^3 e^{-\lambda x}$ | $(-\lambda)^n e^{-\lambda x}$ |

$$F(x) = \frac{1}{x}$$

$$E_{Reciprocal}(Y) = \sum_{i=1}^N \sum_{j=i+1}^N \left(\frac{1}{L_{ij}} - \frac{1}{D_{ij}} - (L_{ij} - D_{ij}) \left(-\frac{1}{D_{ij}^2}\right) \right) = \sum_{i=1}^N \sum_{j=i+1}^N \left(\frac{1}{L_{ij}} - \frac{2}{D_{ij}} + \frac{L_{ij}}{D_{ij}^2} \right) \tag{15}$$

$$F(x) = \frac{1}{x^2}$$

$$E_{InverseQuadratic}(Y) = \sum_{i=1}^N \sum_{j=i+1}^N \left(\frac{1}{L_{ij}^2} - \frac{1}{D_{ij}^2} - (L_{ij} - D_{ij}) \frac{-2}{D_{ij}^3} \right) = \sum_{i=1}^N \sum_{j=i+1}^N \left(\frac{1}{L_{ij}^2} - \frac{3}{D_{ij}^2} + \frac{2L_{ij}}{D_{ij}^3} \right) \tag{16}$$

$$F(x) = e^{-\lambda x}$$

$$E_{BregmanExp}(X) = \sum_{i=1}^N \sum_{j=i+1}^N \left(e^{-\lambda L_{ij}} - e^{-\lambda D_{ij}} + \lambda(L_{ij} - D_{ij})e^{-\lambda D_{ij}} \right) \tag{17}$$

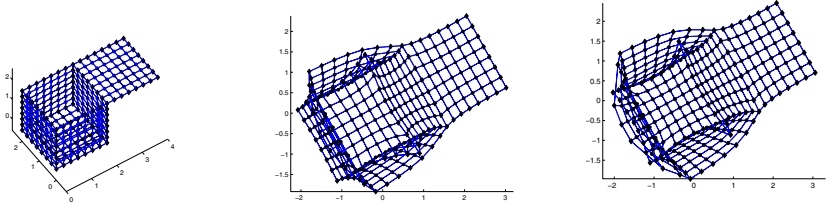
$E_{BregmanExp}$ is an extension of MMDS

$$E_{Exp}(X) = \sum_{i=1}^N \sum_{j=i+1}^N e^{-\lambda D_{ij}} (L_{ij} - D_{ij})^2 \tag{18}$$

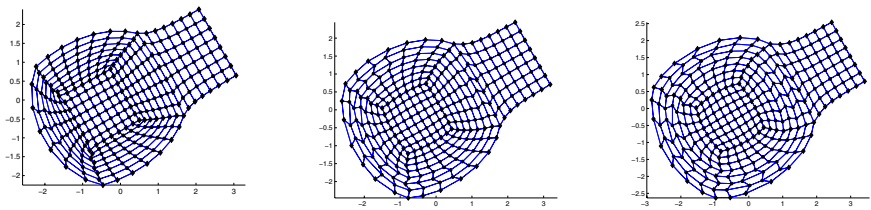
Table 3 summarises Bregmanised stress functions and weights of corresponding MMDS stress functions i.e. that part of the Bregman divergence whose contribution is solely from the second order derivative of the function.

Table 3. Summary of divergences in contrast with corresponding MMDS (eq 11)

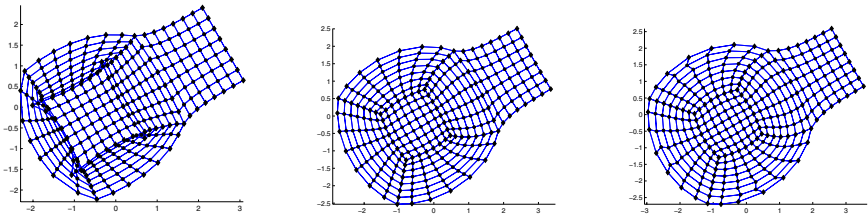
| | Divergence | $F(x)$ | $W(D_{ij})$ | Terms of BMMDS eq 6 |
|-------------|--------------------------|------------------|-----------------------|-------------------------------------------------------------------------------------------|
| FirstGroup | ExtendedSammon eq 12 | $x \log(x)$ | $\frac{1}{D_{ij}}$ | $L_{ij} \log \frac{L_{ij}}{D_{ij}} - L_{ij} + D_{ij}$ |
| | Itakura-Saito eq 14 | $-\log(x)$ | $\frac{1}{D_{ij}^2}$ | $\frac{L_{ij}}{D_{ij}} - \log \frac{L_{ij}}{D_{ij}} - 1$ |
| | Reciprocal eq 15 | $\frac{1}{x}$ | $\frac{1}{D_{ij}^3}$ | $\frac{1}{L_{ij}} - \frac{2}{D_{ij}} + \frac{L_{ij}}{D_{ij}^2}$ |
| | Inverse Quadratic eq 16 | $\frac{1}{x^2}$ | $\frac{1}{D_{ij}^4}$ | $\frac{1}{L_{ij}^2} - \frac{3}{D_{ij}^2} + \frac{2L_{ij}}{D_{ij}^3}$ |
| SecondGroup | Exponential Family eq 17 | $e^{-\lambda x}$ | $e^{-\lambda D_{ij}}$ | $e^{-\lambda L_{ij}} - e^{-\lambda D_{ij}} + \lambda(L_{ij} - D_{ij})e^{-\lambda D_{ij}}$ |



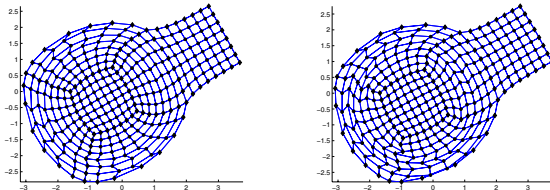
(a) Data set - Open Box (b) Mapped by the Sammon (c) Mapped by the ExtendedSammon mapping



(d) Mapped by Itakura-Saito divergence (e) Mapped data, $F(x) = \frac{1}{x}$ (f) Mapped data, $F(x) = \frac{1}{x^2}$



(g) Mapped data, $F(x) = \exp(-x)$ (h) Mapped data, $F(x) = \exp(-3x)$ (i) Mapped data, $F(x) = \exp(-6x)$



(j) Mapped data, $F(x) = \exp(-10x)$ (k) Mapped data, $F(x) = \exp(-15x)$

Fig. 4. Mappings of the open box data set using the various Bregman divergences

Comparison of strategies. As discussed above the two groups of divergences improve the quality of projection using opposite strategies. The first group works by improving the mapping of short distances, while the second group stretches long distances, which is obviously good for unfolding. In our experiments on the Swiss roll data set, when $t = 2$ for the first group, or $\lambda = 15$ for the second group, the optimisation convergence speed becomes significantly slower.

3.1 Experiment on the Open Box

The open box data set consists of 316 data points. On page 15 of [5], it is stated that “it is connected but neither compact (in contrast with a cube or closed box) nor smooth (there are sharp edges and corners)”.

It is interesting to see in Figure 4 that not all of the distances have the same opportunity to become longer. The result of the first group of divergences is displayed in 4(b) through 4(f). The box opens wider in the mouth, in other words the manifold has been found so that distances between points on opposite sides on the top are longer than on the bottom. Distances between points on the lid remain almost unchanged whereas on the bottom they have contracted. The mouth becomes rounder and rounder, and sharp points and edges are smoothed except on the lid. In a square consisting of four neighbouring data points in the four vertical sides, the horizontal side becomes longer than the vertical side. These results are in contrast to methods such as Curvilinear Component Analysis [3] which tear the box in order to create a two dimensional representation of the data.

4 Conclusion and Future Work

We have shown that linear MDS can be thought of as a Bregman MDS with the special case when the underlying convex function is $F(x) = x^2$ and that the Sammon mapping is a truncated version of a Bregman divergence with the convex function $F(x) = x \log x$. We have shown that the full Bregman mapping of some standard data sets improves upon the mapping provided by the Sammon mapping. We have used our intuition gained from this improvement to develop two groups of mappings based on different underlying convex functions: one group concentrates on shorter distances while the other gives more attention to the longer distances in data space.

Future work will investigate the creation of divergences which merge the properties of our two groups of divergences i.e. which simultaneously pay attention to small and large distances in data space.

References

1. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman divergences. *J. Mach. Learn. Res.* 6, 1705–1749 (2005)
2. Chen, L., Buja, A.: Local Multidimensional Scaling for Nonlinear Dimension Reduction, Graph Drawing and Proximity Analysis. PhD thesis, University of Pennsylvania (2006)

3. Demartines, P., Hault, J.: Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks* 8(1) (1997)
4. Sammon, J.: A nonlinear mapping for data structure analysis. *IEEE Transactions on Computing* 18 (1969)
5. Lee, J.A., Verleysen, M.: *Nonlinear dimensionality reduction* (2007)

Independent Component Analysis Using Bregman Divergences

Xi Wang and Colin Fyfe

School of Computing,
University of the West of Scotland, UK
{xi.wang,colin.fyfe}@uws.ac.uk

Abstract. We review the technique of independent component analysis (ICA) and Stone's criterion for performing an independent component analysis. We then review Bregman divergences and show how they may be applied to Stone's criterion providing a very simple algorithm for performing ICA. We illustrate our method on two very simple data sets.

1 Introduction

Bregman divergences have recently received a great deal of interest recently in terms of clustering and finding unsupervised projections of a data set [2,5,4,3,1]. In this paper we shall investigate Bregman divergences in the context of Independent Component Analysis (ICA): whereas most ICA solutions utilize either kurtosis or entropy considerations (see below), we will concentrate on extensions of Stone's criterion [6].

2 Independent Component Analysis

Independent Component Analysis (ICA) is a method for identifying structure within a data set by ensuring that the components found are as independent as possible. The simplest problem statement starts with a set of d signals, \mathbf{s} , which are linearly mixed by an unknown mixing matrix A . The only assumption we make about the data is that the original signals are independent and that at most one of them is from a Gaussian distribution. The unknown mixing matrix is usually generated randomly in experiments. ICA is the problem of recovering the unknown signals from the mixtures $\mathbf{x} = A\mathbf{s}$. Thus we wish to find a demixing matrix W such that $\mathbf{y} = W\mathbf{x}$ recovers the original signals. There are two indeterminacies inherent in the problem: since we make no assumptions about the mixing matrix and only minimal assumptions about the signals, there is a size indeterminacy. Also we are not in a position to determine the order in which the signals are recovered and so the best we can hope for is that $\exists i : y_i = as_j, \forall j$ and for some constant a .

A common principle for ICA is to make sure all components are as non-Gaussian as possible. According to the central limit theorem, the mixed observations become more Gaussian than any of the independent sources, so we

can just measure the distribution of \mathbf{y} and find a W that maximizes the non-Gaussianity of the elements of \mathbf{y} . A popular way to measure the nongaussianity of a vector is using kurtosis which is the fourth-order cumulant of a random variable.

An alternative criterion is to measure the entropy of the elements of \mathbf{y} . Since a Gaussian random variable has more entropy than any random variable with the same variance, we can use negentropy, the difference between the entropy of the random variable and a Gaussian random variable of the same variance to measure how far we are from a Gaussian distribution.

Stone [6] however created an unusual but very intuitively satisfying criterion for ICA which we review next.

2.1 Stone's Criterion

Stone [6] noted that many signals tend to be locally predictable and thus a criterion which can be used for ICA is to minimise the variance in the local signals. However a matrix W which does this alone is liable to be useless since it will probably tend to zero so that it will output a constant (zero) vector for each sample \mathbf{x} . Thus we require an additional criterion which will move the outputs y away from zero. Stone notes that, to constitute a signal capable of imparting information, the signal must have maximal global variance though it is locally low in variance. Thus he proposes the scalar criterion for each element of \mathbf{y} ,

$$J_{Stone} = \log \left[\frac{\sum_{y_t \in D} (y_t - \bar{y})^2}{\sum_{y_t \in D_t} (y_t - \tilde{y})^2} \right] = \log \left[\frac{V}{U} \right] \quad (1)$$

where y_t is the output, y , at time t , D is the complete data set, D_t is local elements around time t , \bar{y} is the global mean value of y and \tilde{y} is the local mean value. Thus V corresponds to the global variance while U corresponds to the local variance. Maximising J_{Stone} then can be used to identify underlying signals which, while locally smooth and predictable, have high variance globally and therefore informative.

Maximising J_{Stone} then, corresponds to maximising the global variance (thus keeping the solution away from the uninformative constant output), and minimising the local variance (and thus searching for signals which are locally constant).

In practice, both V and U are updated in an online manner so that

$$\begin{aligned} V &= (1 - \lambda_l)V + \lambda_l(y_t - \bar{y})^2 \\ U &= (1 - \lambda_s)U + \lambda_s(y_t - \tilde{y})^2 \end{aligned}$$

where λ_l and λ_s are suitably chosen constants such that λ_l , used for the long-term memory is much smaller than λ_s which is used for the short term memory.

Stone uses gradient ascent in order to maximise J_{Stone} and shows that, in some cases, it can extract the underlying signals from a mixture.

3 Bregman Divergences

Consider a strictly convex function $F : S \rightarrow \mathfrak{R}$ defined on a convex set $S \subset \mathfrak{R}^d$. A Bregman divergence between two elements, p and q , of S is defined to be

$$d_F(p, q) = F(p) - F(q) - (p - q) \cdot \nabla F(q) \tag{2}$$

where the \cdot indicates an inner product and $\nabla F(q)$ is the derivative of F evaluated at q . This can be viewed as the difference between $F(p)$ and its truncated Taylor series expansion around q . Thus it can be used to ‘measure’ the convexity of F : Figure 1 illustrates how the Bregman divergence is the difference between $F(p)$ and the value which would be reached from $F(q)$ with a linear change for $\nabla F(q)$.

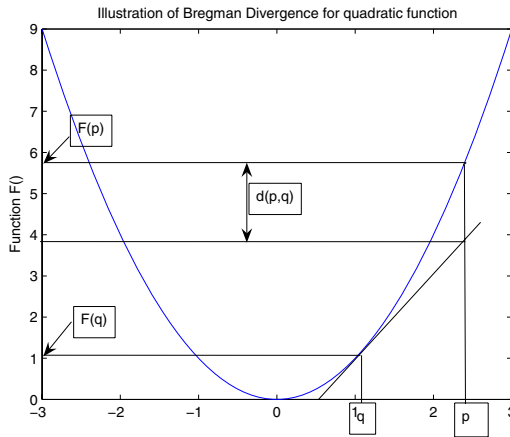


Fig. 1. The divergence is the difference between $F(p)$ and the value of $F(q) + (p - q)\nabla F(q)$

We now give two examples of divergences in common use:

Example 1. The squared Euclidean distance is a special case of the Bregman divergence in which $F(\cdot) = \|\cdot\|^2$

$$\begin{aligned} d_F(\mathbf{x}, \mathbf{y}) &= \|\mathbf{x}\|^2 - \|\mathbf{y}\|^2 - (\mathbf{x} - \mathbf{y}) \cdot \nabla F(\mathbf{y}) \\ &= \|\mathbf{x}\|^2 - \|\mathbf{y}\|^2 - (\mathbf{x} - \mathbf{y}) \cdot 2\mathbf{y} \\ &= \|\mathbf{x} - \mathbf{y}\|^2 \end{aligned}$$

Example 2. The Kullback-Leibler divergence is another special case in which $F(\mathbf{p}) = \sum_{j=1}^d p_j \log p_j$. Consider two discrete probability distributions, \mathbf{p} and \mathbf{q} .

$$d_F(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^d p_j \log_2 p_j - \sum_{j=1}^d q_j \log_2 q_j - (\mathbf{p} - \mathbf{q}) \cdot \nabla F(\mathbf{q})$$

$$\begin{aligned}
 &= \sum_{j=1}^d p_j \log_2 p_j - \sum_{j=1}^d q_j \log_2 q_j \\
 &\quad - \sum_{j=1}^d (p_j - q_j)(\log_2 q_j + \log_2 e) \\
 &= \sum_{j=1}^d p_j \log_2 \frac{p_j}{q_j} - \log_2 e \sum_{j=1}^d (p_j - q_j) \\
 &= \sum_{j=1}^d p_j \log_2 \frac{p_j}{q_j} = K.L.(\mathbf{p} \parallel \mathbf{q})
 \end{aligned}$$

since $\sum_{j=1}^d p_j = \sum_{j=1}^d q_j = 1$. This divergence can be used with general vectors (i.e. not necessarily probability distributions) and then we have the Generalised I-divergence, $d_F(\mathbf{p}, \mathbf{q}) = \sum_{j=1}^d p_j \log \frac{p_j}{q_j} - \sum_{j=1}^d (p_j - q_j)$. Another important divergence is the Itakura-Saito divergence which is based on $F(x) = -\log(x)$. This divergence has been used extensively in signal processing tasks in which the signals show high kurtosis i.e. a small number of very high amplitude values and a much greater number of low amplitude values.

3.1 Properties of Bregman Divergences

Note that, in general, $d_F(\mathbf{p}, \mathbf{q}) \neq d_F(\mathbf{q}, \mathbf{p})$, the exception being the Euclidean distance. We can create symmetric divergences from the asymmetric divergences:

$$\begin{aligned}
 S_F(\mathbf{p}, \mathbf{q}) &= \frac{1}{2}(d_F(\mathbf{p}, \mathbf{q}) + d_F(\mathbf{q}, \mathbf{p})) \\
 &= \frac{1}{2}(\mathbf{p} - \mathbf{q}) \cdot (\nabla F(\mathbf{p}) - \nabla F(\mathbf{q}))
 \end{aligned}$$

This gives us a divergence measured on the space S and its derivative space ∇S .

All Bregman divergences satisfy

Non-negativity $d_F(\mathbf{p}, \mathbf{q}) \geq 0$ with equality if and only if $\mathbf{p} = \mathbf{q}$.

Convexity but only guaranteed in the first parameter.

Linearity $d_{aF_1 + bF_2}(\mathbf{p}, \mathbf{q}) = ad_{F_1}(\mathbf{p}, \mathbf{q}) + bd_{F_2}(\mathbf{p}, \mathbf{q})$

A fuller description of the properties of Bregman divergences can be found in [2].

The fact that we now have a family of divergences to work with raises the problem of which divergence to use with any particular data set or problem. The solution to this is discussed in the next section.

4 The Exponential Family

[2] have shown that there is bijection between a set of Bregman divergences and members of the regular exponential family of probability distributions.

The exponential family of distributions is a very wide family whose members have distributions of the form

$$p_{G,\theta}(\mathbf{x}) = \exp(\mathbf{t}(\mathbf{x}) \cdot \theta - G(\theta)) p_0(\mathbf{t}(\mathbf{x})) \tag{3}$$

where $\mathbf{t}(\mathbf{x})$ is known as the natural statistic, θ is known as the natural parameter and $G(\theta)$ is the cumulant function which defines the exponential family. Examples of the exponential family are

The 1 dimensional Gaussian with unit variance

$$\begin{aligned} p_{G,\theta}(x) &= \frac{1}{\sqrt{2\pi}\sigma} e^{(-\frac{(x-\mu)^2}{2})} \\ &= \frac{e^{-x^2}}{\sqrt{2\pi}} e^{x\mu - \frac{\mu^2}{2}} \end{aligned}$$

So that $t(x) = x$

$$\theta = \mu$$

$$\text{and } G(\theta) = \frac{\theta^2}{2} = \frac{\mu^2}{2}$$

The one dimensional Bernoulli

$$p_{G,\theta}(x) = p^x(1-p)^{1-x} = 1e^{x\theta - \log(1+e^\theta)}$$

$$\text{where } \theta = \log\left(\frac{p}{1-p}\right)$$

and $t(x) = x$

$$\text{and } G(\theta) = \log(1 + e^\theta)$$

Other well known members of this family are the multinomial, beta, Dirichlet, Poisson, Laplace, gamma and Rayleigh distributions. In the remainder of this paper, we consider only regular exponential families in which $\mathbf{t}(\mathbf{x}) = \mathbf{x}$.

We define the expectation of \mathbf{X} with respect to $p_{G,\theta}$ to be

$$\mu = E_{p_{G,\theta}}[X] = \int_{\mathbb{R}^d} \mathbf{x} p_{G,\theta}(\mathbf{x}) d\mathbf{x} \tag{4}$$

It can be shown [2] that there is a bijection between the set of expected values, μ , and the set of natural parameters, θ . In fact, let d_F be the Bregman divergence corresponding to the distribution, $p_{G,\theta}$. Then let $g(\cdot) = \nabla G$ and let $f = \nabla F$. Then $\mu = g(\theta)$ and $\theta = f(\mu)$, which is readily verified for the distributions above.

Consider a member of the regular exponential family with known cumulant function, $G(\theta)$. Then $G(\cdot)$ is a closed convex function. Define its conjugate function as

$$F(\mathbf{x}) = \sup_{\theta} \{\mathbf{x} \cdot \theta - G(\theta)\} \tag{5}$$

Then there is an unique θ^* which attains the supremum and $F(\cdot)$ is also a convex function. If the domain of F is S and the domain of G is Θ , then (S, F) is

the Legendre dual of (Θ, G) . In particular, there exists a θ such that $F(\mu) = \mu.\theta - G(\theta)$. Differentiating and setting the derivative to 0, we see that $g(\theta) = \mu$ and $f(\mu) = \theta$; then since $G(\cdot)$ is strictly convex, $F(\cdot)$ is too and so can be used to define a Bregman divergence. Consider two members of an exponential family with natural parameters, θ_1 and θ_2 , and expectations, μ_1 and μ_2 . Then

$$\begin{aligned} d_G(\theta_1, \theta_2) &= G(\theta_1) - G(\theta_2) - (\theta_1 - \theta_2).g(\theta_2) \\ &= \mu_1.\theta_1 - F(\mu_1) - \mu_2.\theta_2 + F(\mu_2) \\ &\quad - (\theta_1 - \theta_2).\mu_2 \\ &= F(\mu_2) - F(\mu_1) - (\mu_2 - \mu_1).\theta_1 \\ &= F(\mu_2) - F(\mu_1) - (\mu_2 - \mu_1).f(\mu_1) \\ &= d_F(\mu_2, \mu_1) \end{aligned}$$

Thus minimising the Bregman divergence with respect to the cumulant function between the natural parameters is equivalent to minimising the Bregman divergence with respect to the dual function (but in the opposite direction) between the expectations. Also

$$\begin{aligned} \log p_{G,\theta}(x) &= \log p_0(\mathbf{x}) + \mathbf{x}.\theta - G(\theta) \\ &= \log p_0(\mathbf{x}) + F(\mathbf{x}) \\ &\quad - \{F(\mathbf{x}) + G(\theta) - \mathbf{x}.\theta\} \\ &= \log p_0(\mathbf{x}) + F(\mathbf{x}) \\ &\quad - \{F(\mathbf{x}) - F(\mu) + \mu.\theta - \mathbf{x}.\theta\} \\ &= \log p_0(\mathbf{x}) + F(\mathbf{x}) - \{F(\mathbf{x}) - F(\mu) \\ &\quad + \mu.f(\mu) - \mathbf{x}.f(\mu)\} \\ &= \log p_0(\mathbf{x}) + F(\mathbf{x}) - d_F(\mathbf{x}, \mu) \end{aligned}$$

Thus maximising the likelihood of a data set is equivalent to minimising the associated Bregman divergence between the mean of the distribution and the data.

In practical terms, we might fit a particular member of the exponential family to a data set which means we have determined the cumulant function, $G(\cdot)$. We then identify the dual function, $F(\cdot)$, based on which we can find the Bregman divergence $d_F(\cdot)$ knowing that minimising the Bregman divergence between the mean of the distribution and its natural statistics maximises the log likelihood of the distribution under this probability density function.

5 Bregmanising Stone’s Method

Recall that Stone’s criterion is

$$J_{Stone} = \log \left[\frac{\sum_{y_t \in D} (y_t - \bar{y})^2}{\sum_{y_t \in D_t} (y_t - \tilde{y})^2} \right] = \log \left[\frac{V}{U} \right] \tag{6}$$

We can Bregmanise this but first we must note that the Bregman divergence is not symmetric and so we may get different results depending on where the averages appear in the criteria. Thus we have

$$J_{RBStone} = \log \left[\frac{\sum_{y_t \in D} (d_F(y_t, \bar{y}))}{\sum_{y_t \in D_t} (d_F(y_t, \tilde{y}))} \right] = \log \left[\frac{V_R}{U_R} \right] \tag{7}$$

where we have used $J_{RBStone}$ as shorthand for the right Bregman divergence using Stone’s method, and correspondingly

$$J_{LBStone} = \log \left[\frac{\sum_{y_t \in D} (d_F(\bar{y}, y_t))}{\sum_{y_t \in D_t} (d_F(\tilde{y}, y_t))} \right] = \log \left[\frac{V_L}{U_L} \right] \tag{8}$$

when the averages are in the left position in the divergences. It would be possible to consider combinations of V_R, U_L and V_L, U_R but we find it difficult to find a rationale for these.

Thus, for example if we are using the Itakura Saito divergence and so $F(\cdot) = -\log(\cdot)$, then the criterion with the average in the right position would be J_{RIS} (Right Itakura Saito). This gives

$$V_R^{IS} = \sum_{y_t \in D} -\log(y_t) + \log(\bar{y}) + \frac{(y_t - \bar{y})}{\bar{y}}$$

$$U_R^{IS} = \sum_{y_t \in D_t} -\log(y_t) + \log(\tilde{y}) + \frac{(y_t - \tilde{y})}{\tilde{y}}$$

which gives a parameter update (for the presentation of a single input) of

$$\Delta w_i = \alpha \left[\left(-\frac{x_i}{y} + \frac{\bar{x}_i}{\bar{y}} + \frac{(\bar{y}x_i - y\bar{x}_i)}{\bar{y}^2} \right) \frac{1}{V_R^{IS}} - \left(-\frac{x_i}{y} + \frac{\tilde{x}_i}{\tilde{y}} + \frac{(\tilde{y}x_i - y\tilde{x}_i)}{\tilde{y}^2} \right) \frac{1}{U_R^{IS}} \right] \tag{9}$$

where α is a learning rate and we have removed the subscript t for clarity.

In practice, we robustify the algorithm in that we add a small constant to the denominator of every fraction to ensure we are not dividing by 0.

Similarly if we consider the GI divergence, then we have $F(x) = x \log x$. The right divergence gives us

$$V_R^{GI} = y \log(|y|) - \bar{y} \log(|\bar{y}|) + (1 + \log(|\bar{y}|))(y - \bar{y})$$

$$U_R^{GI} = y \log(|y|) - \tilde{y} \log(|\tilde{y}|) + (1 + \log(|\tilde{y}|))(y - \tilde{y})$$

and the learning rule becomes

$$\Delta w_i = \alpha \left[\left(x_i \log \left(\left| \frac{y}{\bar{y}} \right| \right) + \bar{x}_i \frac{\bar{y} - y}{\bar{y}} \right) \frac{1}{V_R^{GI}} - \left(x_i \log \left(\left| \frac{y}{\tilde{y}} \right| \right) + \tilde{x}_i \frac{\tilde{y} - y}{\tilde{y}} \right) \frac{1}{U_R^{GI}} \right] \tag{10}$$

6 Simulations

6.1 Artificial Data

We begin with a simple demixing exercise on artificial data. We construct 3 signals each of 1000 samples,

$$\begin{aligned} s_1(t) &= \cos\left(\frac{\pi t}{10}\right) + \epsilon \\ s_2(t) &= \cos\left(\frac{\pi t}{73}\right) + \epsilon \\ s_3(t) &\sim U(0.5, 1.0) \end{aligned}$$

where $s_i(t)$ denotes the i^{th} signal at time t and $U(a, b)$ denotes a uniform distribution between a and b . ϵ is simply an offset to ensure that at no stage were we taking the logarithm of a negative number.

The random mixing matrix, A was

$$\begin{bmatrix} 0.751442 & 0.903463 & 0.228328 \\ 0.955989 & 0.038796 & 0.020169 \\ 0.612619 & 0.395016 & 0.108775 \end{bmatrix} \quad (11)$$

The three mixtures are shown in the first three diagrams of Figure 2 and the final output when we use the right Itakura Saito divergence (corresponding to J_{RIS}) is shown in the bottom right. We see that the low frequency sinusoid has been recovered: the correlation between the output found and the second signal was 0.999. Note that this was with the signals corrupted with noise from a uniform distribution (a platykurtotic distribution) which is far from the best noise distribution for a criterion which is optimal for leptokurtotic noise. We also show in Figure 3 the corresponding values of \bar{y} and \tilde{y} : we see that \tilde{y} gives a very smooth (indeed exact) representation of the original signal. For this result, we used $\lambda_l = 0.00002$, $\lambda_s = 0.1$, $\alpha = 0.0001$.

6.2 Image Identification

We now added together linearly 3 images, each 512×512 : they were the well-known ‘‘Lena’’, a picture of water and a herringbone texture. The original images are shown in Figure 4. The random mixing matrix is

$$\begin{bmatrix} 0.468080 & 0.301993 & 0.039809 \\ 0.488997 & 0.722717 & 0.856870 \\ 0.921170 & 0.639605 & 0.345576 \end{bmatrix} \quad (12)$$

We used a right Itakura-Saito divergence with $\alpha = 0.0001$ decaying to 0 during the experiment (3000 iterations over the whole data set), $\lambda_1 = 0.0002$ and $\lambda_s = 0.2$.

The final output has correlations 0.991, 0.063 and 0.022 respectively with each of the original images. ‘‘Lena’’ is clearly identified. This is perhaps not surprising since ‘‘Lena’’ is by far the most slowly changing image of the three except of course round the eyes etc.

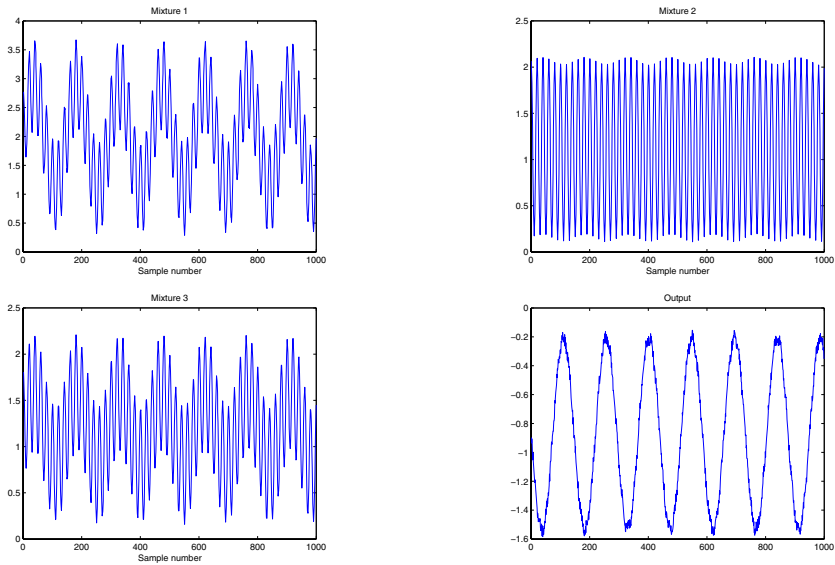


Fig. 2. The top 2 diagrams and the left diagram on the bottom show the three mixtures, the bottom right diagram shows the output signal when we use the criterion J_{RIS} which has a correlation of 0.999 with the second (low frequency) sinusoid

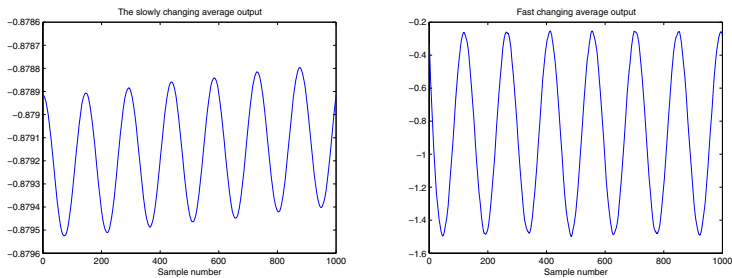


Fig. 3. The slowly changing value \bar{y} (left) varies within a very small range, while the fast changing value, \tilde{y} has clearly identified the second signal



Fig. 4. From left to right: water, lena and herringbone

7 Conclusion

We have taken Stone's criterion for independent component analysis and used Bregman divergences to create somewhat different algorithms. The different divergences are known to be optimal in the context of Bregmanising K-means clustering. We are now engaged in empirically testing whether these divergences are optimal for independent component analysis using Stone's criterion. Also Stone used the same criterion for the discovery of disparity in visual images, interestingly where the disparity changes gradually over the image. We also are investigating whether the Bregmanised versions of Stone's criterion can be used in a similar fashion.

References

1. Azoury, K.S., Warmouth, M.K.: Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning* (43), 211–246 (2001)
2. Banerjee, A., Meruga, S., Dhillon, I., Ghosh, J.: Clustering with Bregman divergences. *Journal of Machine Learning Research* 6, 1705–1749 (2005)
3. Collins, M., Dasgupta, S., Shapire, R.E.: A generalization of principal component analysis to the exponential family. In: *Nips14* (2002)
4. Neilsen, F., Boissonnat, J.-D., Nock, R.: Bregman Voronoi diagrams: Properties, algorithms and applications (2007) (submitted)
5. Neilsen, F., Boissonnat, J.-D., Nock, R.: On Bregman Voronoi diagrams. In: *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 746–755 (2007)
6. Stone, J.V.: Blind source separation using temporal predictability. *Neural Computation* 13(7), 1559–1574 (2001)

Novel Method for Feature-Set Ranking Applied to Physical Activity Recognition

Oresti Baños, Héctor Pomares, and Ignacio Rojas

Department of Computer Architecture and Computer Technology, University of Granada,
C/Periodista Daniel Saucedo Aranda s/n E-18071, Granada, Spain
oresti@correo.ugr.es, {hector,irojas}@ugr.es

Abstract. Considerable attention is recently being paid in e-health and e-monitoring to the recognition of motion, postures and physical exercises from signal activity analysis. Most works are based on knowledge extraction using features which permit to make decisions about the activity realized, being feature selection the most critical stage. Feature selection procedures based on wrapper methods or ‘branch and bound’ are highly computationally expensive. In this paper, we propose an alternative filter method using a feature-set ranking via a couple of two statistical criteria, which achieves remarkable accuracy rates in the classification process.

Keywords: Activity Recognition, Feature Selection, Ranking.

1 Introduction

Daily physical activity recognition is a very important task that is currently being applied in several fields such as e-health, robotics, sports, videogames industry, among others [4,7,9,10,11]. The primary difficulty consists in designing a system whose reliability is independent of who is carrying out the exercise and the particular activity execution style. Besides, the complexity is increased by distortion elements related to system monitoring and processing, along with the random character of the execution.

In the literature, most studies performed are based on supervised laboratory data. Nevertheless, the apparently good recognition results on supervised data that some works achieve cannot be extrapolated to unsupervised (semi-naturalistic) data [2,12]. In this paper we propose an automatic methodology to extract a set of the most important features to be used in activity recognition. One of the most important characteristic of the method proposed is that we do not provide a rank order for every individual feature but for every set of features, allowing for the synergical utility of several features when considered together at the same time.

The rest of the paper is organized as follows: In Section 2 we make a brief summary of the activity recognition process. Next, we present the rank-based feature-set selection methodology developed, describing the fundamentals of this method and the algorithm's main steps. Finally we evaluate the performance of the method for a specific example and we compare the accuracy results with related previous works.

2 Activity Recognition

Our experimental setup starts from a signal set corresponding to acceleration values measured by a group of sensors located in strategic different parts of the body (hip, wrist, arm, ankle, thigh), for several daily activities (see Figure 1). This philosophy of work can easily be generalized to other studies related to activity recognition from a set of features.

The initial information provided by the sensors has some artefacts and noise associated to the acquisition data process. A low pass and high pass filtering is normally used to remove these irregularities.

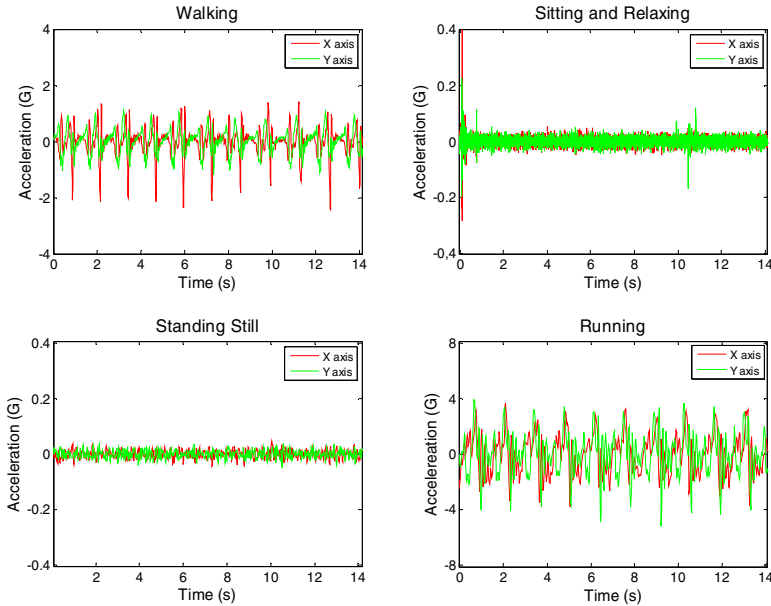


Fig. 1. Signals corresponding to 4 usual daily physical activities. It's relevant the signal pattern form similarity between “walking” and “running”, and “sitting and relaxing” and “standing still” respectively.

Subsequently we generate a parameter set made up of 861 features corresponding to a combination of statistical functions such as mean, kurtosis, mode, variance, etc., and magnitudes obtained from a domain transformation of the original data such as energy spectral density, spectral coherence or wavelet decomposition, among others. These features are evaluated over the complete signal, although other alternatives based on windowing and sub-segmentation signal feature extraction could also be tested.

In this stage, we now must rely on a feature selection process that has the responsibility of deciding which features or magnitudes are the most important ones to decide the kind of activity the person is carrying out. In the next section we describe the method we have designed to accomplish this task.

3 Proposed Ranking Technique Based on Discrimination and Robustness

Obtaining a specific group of variables from a big initial set is not a trivial task, because the number of possible feature combinations is huge. In our experimental setup the sample space is represented by $n = 861$ features, so brute force techniques like ‘branch and bound’ ($O(2^n)$ convergence $\rightarrow 2^{861} \approx 1.5 \times 10^{259}$ possible permutations) or wrapper methods are impractical. In this section, we present an alternative method based on the concepts of discrimination and robustness for a complete set of features.

Let us define the *sample range* of a class as the set of values included between the maximum and the minimum value (both inclusive) that a feature or variable takes for this class. Every circle showed in the example of figure 2 represents a sample corresponding to the feature value calculated over the data from a concrete subject and class. We represent for every class the sample range using a right brace sign with the same class tonality.

Given a group of samples (associated to every class) we rank its discriminant capability with respect to that class through the overlapping probability between this class and the others. This is calculated computing the number of samples from the analyzed class which are inside of the sample range defined by the others. For N classes and M samples for each class (let us suppose that this number is independent of the class), we define the overlapping probability of a set of samples as follows:

$$p(k) = \frac{1}{N} \sum_{n \neq k}^N \frac{m(k,n)}{M} \quad (1)$$

with $m(k,n)$ being the number of samples from the class k inside the sample range of class n .

In order to make this more understandable, for the example given in figure 2 ($N = 4$, $M = 8$), the overlapping probability for the class ‘‘running’’ is $p = \frac{1}{4}(3/8+0/8+0/8) = 0.094$, since there are 3 samples from the class ‘‘running’’ in the data range defined for class ‘‘walking’’, and 0 in the rest of classes. Consequently, this feature permits to discriminate a priori the activity *running* from the activity *standing still* or *sitting and relaxing*, but it could be mistaken with an approximately 9% probability with *walking*.

We now carry out a thresholding process which allows us to define the feature analyzed as discriminative or not. This *overlapping* threshold takes values from 0 (the most restrictive, for cases with no overlapping between classes) to 1 (the most relaxed, when every sample from a class is inside the others). In general for a specific feature, if the analyzed class exceeds the threshold, the feature will be considered as no discriminant for this class.

Apart from the discriminant capacity of a feature or a set of features, a second characteristic is now defined which takes into account the usability of this set of features in different information contexts or sources. For instance, a specific measure taken from, let's say, the ankle accelerometer can be very discriminative to distinguish between the activities *walking* and *standing still*, but this very same measure may not be that reliable when taken from the thigh accelerometer. There may be some measures with the same discriminant capability between those activities which are not so

dependent of the exact location of the sensor or, at least, which are still reliable when taken from a bigger number of sensors. We will denote this measure as the robustness criterion of a set of features. In short, discriminant capacity says how useful a motion feature is in general, and robustness is how this depends on where the sensor is.

Combining both criteria we obtain a quality ranking procedure capable of grouping features in different stages. For the sake of simplicity, let us suppose a recognition system with 4 classes and 5 sources; features will be classified in groups defining a ranking (see table 1). For instance, features that discriminate 4 classes in every source will be added to group #1 (the best). Group #14 will be completed with features that classify 2 classes (the same) in 3 sources at least. This example is extensible to any classes and sources.

4 Results

To evaluate the effectiveness of the ranking method developed, we use a signal database¹ corresponding to the data monitored by 5 biaxial accelerometers (hip, wrist, arm, ankle and thigh) for 4 activities (introduced in figure 1) in laboratory (supervised) and semi-naturalistic (unsupervised) environments.

Most remarkable features (set #1 and #2 primarily) for supervised and unsupervised data are geometric mean for amplitude signal, autocorrelation and some wavelets coefficients obtained through a 5-level Daubechies decomposition. This together with a classification strategy based on C4.5 decision tree permits to achieve an accuracy rate close to 99% ($98.92\% \pm 1.08\%$) for laboratory data and 95% ($95.05\% \pm 1.20\%$) for semi-naturalistic data. We used a cross validation method for training and

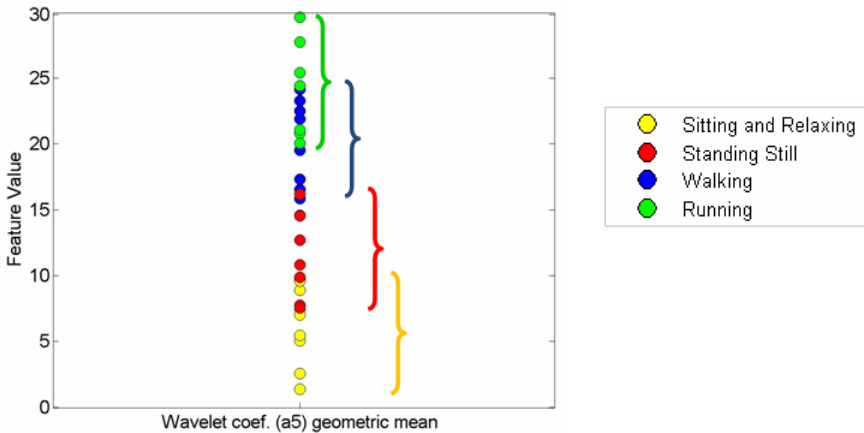


Fig. 2. Feature values extracted for the ankle accelerometer. Every circle represents the value (sample) of the feature “decomposition wavelet coefficients a5 geometric mean” for a specific subject (8 in all), identifying with different tonalities the belonging class.

¹ Database facilitated in [2] by Prof. Stephen Intille (Massachussets Institute of Technology).

Table 1. Example for 4 classes and 5 sources of quality feature set (ranking) based on discriminant (number of activities discriminated, first column) and robustness (number of motion sensor where the feature is discriminant, second column) criterions

| Discriminant capacity | Robustness | Quality group |
|-----------------------|------------|---------------|
| 4 | 5 | #1 |
| 4 | 4 | #2 |
| 4 | 3 | #3 |
| 4 | 2 | #4 |
| 4 | 1 | #5 |
| 3 | 5 | #6 |
| 3 | 4 | #7 |
| 3 | 3 | #8 |
| 3 | 2 | #9 |
| 3 | 1 | #10 |
| 2 | 5 | #11 |
| 2 | 4 | #12 |
| 2 | 3 | #13 |
| 2 | 2 | #14 |
| 2 | 1 | #15 |
| 1 | 5 | #16 |
| 1 | 4 | #17 |
| 1 | 3 | #18 |
| 1 | 2 | #19 |
| 1 | 1 | #20 |
| 0 | - | #21 |

testing. These results are quite good due to the relative parallelism between probability model used in feature selection defined and the entropy-based model used in decision tree.

Although a strict comparison with other studies cannot be made since the data and the number of classes may differ, in [9] a 83-90% classification accuracy was reached for laboratory conditions, 92.85%-95.91% in [8] (also for lab conditions), 89% in [2] for supervised and unsupervised, or 93% and 89% on recent works ([3] and [5] respectively) for semi-naturalistic data.

5 Conclusions

In this work we have very briefly shown a direct application of ranking selection methods used on daily physical activity automatic recognition. An efficient classification method requires a productive and limited feature set, being necessary a selection process since the initial set is quite huge. We have defined a feature selector based on statistical discrimination and robustness criteria, focused on low computational time and resources, defining a real alternative to other selection processes.

For future work, we aim to make a time-based comparison to traditional features selectors [6, 13, 14].

Acknowledgments. This work has been supported by the Spanish CICYT Project TIN2007-60587, Junta de Andalucía Project P07-TIC-02768 and the CENIT project AmIVital, of the "Centro para el Desarrollo Tecnológico Industrial" (CDTI- Spain). We want to express our gratitude to Prof. Stephen S. Intille, Technology Director of the House_n Consortium in the MIT Department of Architecture for the experimental data provided.

References

1. Baek, J., Lee, G., Park, W., Yun, B.J.: Accelerometer Signal Processing for User Activity Detection. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) KES 2004. LNCS (LNAI), vol. 3215, pp. 1611–3349. Springer, Heidelberg (2004)
2. Bao, L., Intille, S.S.: Activity Recognition from User-Annotated Acceleration Data. In: Ferscha, A., Mattern, F. (eds.) Pervasives 2004. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004)
3. Bonomi, A.G., Goris, A.H.C., Yin, B., Westerterp, K.R.: Detection of Type, Duration, and Intensity of Physical Activity Using an Accelerometer. *Medicine & Science in Sports & Exercise* 41, 1770–1777 (2009)
4. Crampton, N., Fox, K., Johnston, H., Whitehead, A.: Dance, Dance Evolution: Accelerometer Sensor Networks as Input to Video Games. In: IEEE International Workshop on Haptic, Audio and Visual Environments and Games, pp. 107–112 (2007)
5. Ermes, M., Pärkka, J., Mantyjarvi, J., Korhonen, I.: Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Trans. Inf. Technol. Biomed.* 12, 20–26 (2008)
6. Kohavi, R., Sommereld, D.: Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. In: First International Conference on Knowledge Discovery and Data Mining (1995)
7. Koskimaki, H., Huikari, V., Siirtola, P., Laurinen, P., Roning, J.: Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. In: 17th Mediterranean Conference on Control and Automation, pp. 401–405 (2009)
8. Lee, S.W., Mase, K.: Activity and location recognition using wearable sensors. *IEEE Pervasive Computing* 1, 24–32 (2002)
9. Mantyjarvi, J., Himberg, J., Seppanen, T.: Recognizing human motion with multiple acceleration sensors. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 747–752 (2001)
10. McCue1, M., Hodgins, J., Bargteil, A.: Telerehabilitation in Employment/Community Supports Using Videobased. In: Activity Recognition. RERC on Telerehabilitation (2008)
11. Munguia, E., Intille, S.S., Haskell, W., Larson, K., Wright, J., King, A., Friedman, R.: Real-Time Recognition of Physical Activities and Their Intensities Using Wireless Accelerometers and a Heart Rate Monitor. In: Proceedings of the 2007 11th IEEE International Symposium on Wearable Computers, pp. 1–4 (2007)
12. Ravi, N., Dandekar, N., Mysore, P., Littman, M.L.: Activity Recognition from Accelerometer Data. In: Proceedings of the 17th conference on Innovative applications of artificial intelligence, pp. 1541–1546 (2005)
13. Song, L., Smola, A., Gretton, A., Borgwardt, K.M., Bedo, J.: Supervised feature selection via dependence estimation. In: Proceedings of the 24th international conference on Machine learning, pp. 823–830 (2007)
14. Xu, Z., Jin, R., Ye, J., Lyu, M.R., King, I.: Non-monotonic feature selection. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1145–1152 (2009)

Time Space Tradeoffs in GA Based Feature Selection for Workload Characterization

Dan E. Tamir¹, Clara Novoa², and Daniel Lowell¹

¹ Department of Computer Science

² Ingram School of Engineering

Texas State University

San Marcos Texas, 78666 USA

{dt19, cn17, dlowell}@txstate.edu

Abstract. This paper reports the results of a research effort that explores time/space tradeoffs inherent to genetic algorithms (GA). The study analyzes redundancy in the GA search space and lays out a schema for efficient utilization of record keeping in the form of a cache to minimize redundancy. The application used for evaluation of the record keeping procedure is feature selection for computer workload characterization. The experimental results demonstrate the utility of record keeping in the GA domain, and show a significant reduction in execution time with virtually the same solution quality.

Keywords: Heuristic search, Genetic Algorithms, Feature Selection, Workload Characterization, Cache Design.

1 Introduction

One form of computer workload characterization involves clustering of fixed size slices of a trace of an application code obtained from a high level instruction set simulator, and identifying prototype slices [1, 2]. Each trace-slice is represented by a set of features such as the number of arithmetic instructions in the slice, the number of memory references, and the number of register's bits altered. Prototype slices are analyzed through a low level software simulator and the results are used to estimate the performance of the entire code [1, 2]. Numerous architecture and micro-architecture features can be extracted from the trace. Nevertheless, it is desired to identify an optimal feature sub-set in order to enable cost effective characterization.

The current work stems from a research project where several methods for feature selection including Genetic Algorithms (GA) have been considered, and GA has been found to be the most promising method [3-6].

This research explores time/space tradeoffs that are inherent to GAs and can reduce the time complexity of the GA procedure and/or improve the quality of its feature selection results. Time/space trade-offs are explored using a record keeping mechanism in the form of a cache that maintains a partial list of chromosomes encountered throughout the GA execution. Experiments performed show that record keeping significantly reduces the execution time achieving virtually the same solution quality.

Alternatively, if the execution time is fixed the record keeping method might result in a better solution. Results of this research are not limited to workload characterization or feature selection. In fact, further ongoing research effort targets using time/space trade-offs to improve GA performance for feature selection problems in other domains, to explore the utility of the approach in other search spaces such as the Traveling Salesman Problem (TSP) search space, and to improve other heuristic search procedures.

The rest of the paper is organized as follows: Section 2 discusses the topic of feature selection. Section 3 describes record keeping mechanisms. Section 4 presents experiment results as well as result evaluation. Finally, section 5 provides conclusions and directions for future research.

2 Feature Selection

Automatic, non-supervised, pattern recognition and classification is a fundamental Artificial intelligence (AI) problem. A typical approach to this problem is to extract a selected set of features from each pattern under consideration and compare the features of unknown patterns with the features of prototype patterns. Often, many features can be extracted, and different features have a different contribution to classification accuracy. Moreover, the computational complexity of the classification process is affected by the number of features used. Hence, it may be desirable to select an optimal subset of features that would enable cost effective classification; where the optimality criteria relates to the discriminatory properties of the selected subset. Assume that M features can be extracted from the patterns. Hence, there are 2^M combinations of feature subsets. For small values of M it might be possible to perform exhaustive search for the optimal subset of features by enumerating all the possible 2^M combinations and checking each combination against the optimality criteria. Nevertheless, the enumeration is exponential with respect to M , and for large values of M exhaustive search becomes prohibitively computationally expensive. In fact, the feature selection problem (FSP) is known to be an NP-hard problem [7].

In many cases, computational complexity considerations and implementation constraints dictate the desired number of features in the selected subset (say n). Under these constraints the FSP boils down to finding the optimal subset of n features from a superset of M features. This statement of the FSP entails selecting one of the $\binom{M}{n}$ combinations of features according to an optimality criterion. Despite the cardinality reduction from 2^M to $\binom{M}{n}$, for a fixed n , the problem is still exponential in M and is NP complete [7].

The FSP can be stated as a combinatorial optimization problem relevant to the areas of pattern recognition, statistics, and machine learning [8]. Feature selection has been applied to diverse practical problems such as medical diagnosis, spam detection, data mining, and data compression. It is used to detect major explanatory variables [9], reduce measurement costs [10], increase the speed of the classification tasks, facilitate accurate inferences by reducing the influence of noise [11, 12], and reduce storage needs as well as bandwidth consumption [13].

Heuristic search techniques for solving the FSP include suboptimal branch and bound, sequential forward/backward selection, and plus-1 take-away-r selection [14, 15]. In addition, there is a considerable number of studies related to meta-heuristic approaches such as genetic algorithms [4], memetic algorithms [16], tabu search [10], particle swarm [17], and ant-colony systems [12].

Siedlecki and Sklansky use GAs for solving a version of the FSP that searches for the minimal feature subset for which the classification quality is above a pre-defined threshold [5]. The authors compare sequential search, branch and bound (BB), and GA. The experiments are done with simulated data and with real data having 150-300 features. Their results show that GA outperforms all the researched methods. It is more efficient than BB, and visits the search space in a more complete way than sequential search.

After thorough evaluation of the literature, in specific the research reported in [4-6], and following a set of experiments with several heuristics, GA was selected as the main platform for the FSP solution in the workload characterization project. In addition, GA was selected to explore time/space trade-offs using cache memory.

3 Record Keeping Mechanisms

Many heuristic algorithms present redundancy; as it might be possible that the same candidate solution is evaluated more than one time. Formally, redundancy in a heuristic search routine is defined to be the ratio of the total number of states explored to the number of distinct states explored. That is: $Redundancy = \frac{\text{number of states explored}}{\text{number of distinct states}}$. The redundancy of a problem can be used to infer an upper bound on the speedup that can be obtained through record keeping.

In the context of GA redundancy can be significant. In many variants of the algorithm, a large portion of the current population of chromosomes has already been evaluated in previous iterations of the algorithm. This brings an interesting question, which is not sufficiently addressed in the literature. The question relates to the type of records that could be kept throughout the GA session and the best policy to save these records in order to minimize redundancy and improve execution efficiency.

Three memory models are used to analyze redundancy and the potential of record keeping: the minimum memory model, the infinite memory model, and the cache model. The **minimum memory model** (also referred to as the **no cache model**) assumes that there is no cache or dedicated memory that can be used to record states previously encountered. Hence, no record keeping, beyond the record keeping specifically implied by the algorithm (e.g., maintaining the elite list in GA), is performed and the algorithm is implemented with the minimum amount of memory possible. **Caching** is described later in the section. The **infinite memory model** is a theoretical model that assumes that there is no bound on the memory used for record keeping, and assumes that there is enough memory to store all the distinct states (chromosomes) encountered during the search. The model is used to analyze the time/space tradeoffs. For the FSP and many other heuristic search procedures, recording all the states encountered would transform the problem from NP-complete to NP-space. This is, of course, not desirable. Practically, in this study, the experiments have a fixed number of generations and use dedicated memory. The memory is large enough to

store every distinct chromosome generated by the minimal memory model. Hence, it can be considered as a model of “infinite” dedicated memory or an infinite table for record keeping of all the distinct chromosomes.

3.1 Caching in Computer Architecture

Computer memory systems exploit two empirical principles: “locality of reference” as well as “small is fast,” and implement a memory hierarchy. Generally, under this scheme, the locality of the current central processing unit (CPU) reference word is stored in a small and fast cache [18]. Related to the concepts of locality of reference are the terms of hit and miss and hit/miss rate. A hit occurs when a CPU reference-word is located in the cache. A miss occurs when the word is not in the cache and has to be brought from memory. A hit may incur a minor overhead whereas a miss entails a significant penalty that includes evicting a block from the cache and replacing this block with the block that contains the current locality of reference. Three basic replacement policies and several dynamic / adaptive combinations of the three are generally considered [18]. The first policy is ‘random replacement’. It randomly chooses the block to be evicted. Next, a recency-based method, evicts the least recently used block. This method mainly exploits temporal locality. Finally, a frequency-based method evicts the least frequently used block, thereby exploiting spatial locality. This study implements and tests these three methods.

There are three main cache organization methods that relate to the way addresses in main memory are mapped to cache address: direct mapping, set associative mapping, and fully associative mapping [18]. Direct mapping does not enable efficient replacement policies and associative memory is quite expensive. Hence, in practice, set associative mapping which enables implementing replacement policies with a reasonable associativity of 4 to 16 is used. For the experiments performed in this research, set associative mapping is used.

3.2 Software Caching

Several researchers including Hertel and Pitassi [19] as well as Allen and Darwiche [20] have studied time/space trade-offs in the context of heuristic search. They found that time requirements could be significantly reduced through record keeping. They referred to their method as caching. Nevertheless, their cache is static; they do not consider cache replacement policies, or cache organization issues. Moreover, their findings do not pertain to GA.

Aggarwal investigates the technique of software caching for memory intensive applications that perform searches or sorted insertions [21]. He obtains reductions of up to 30% in computational time due to caching. However, Aggarwal’s implementation is not extended to heuristic optimization problems such as the one considered in this paper.

Karhi and Tamir demonstrate that iterative hill climbing (IHC); can get better performance by exploiting time/space tradeoffs emerging from saving intermediate results of the search [22]. The IHC algorithm with record keeping method that is analogous to the mechanism of cache is used to solve a traveling salesman problem (TSP).

Santos et al, investigate cache diversity in GAs. The cache is used to store partial results of the chromosome evaluation function [23]. Nevertheless, they assume that the chromosome evaluation function can be decomposed into small units that represent the evaluation of parts of the chromosome and then recomposed based on mutations and crossovers of these parts. Moreover, despite referring to their record keeping as cache, the record keeping mechanism does not include provisions for replacement policies and can actually be considered as infinite dedicated memory. This limits their approach to small problems and to problems where the fitness function computation can be decomposed and recomposed.

The main thrust of our research is that redundancy that is equivalent to the locality of reference in computer systems exists in many heuristic search spaces and algorithms. In GA, the locality is due to the fact that the best solutions are retained and used for constructing future solutions. Due to the evidence for redundancy and locality of reference obtained from previous research [24] it has been decided to explore the time/space tradeoffs associated with record keeping in the form cache in the context of the GA. To the best of our knowledge, the research reported in this paper is the first comprehensive and scalable study of time/space-offs due to caching within the context of GA.

4 Experiments and Results

This section presents the GA implementation in the FSP domain with and without caching. Next, a set of experiments along with a summary of results is presented followed by results analysis.

4.1 Experimental Setup

Raw data provided by a semiconductor company interested in this research is used to evaluate the performance of the proposed GA with caching. The data contains a trace of four computer benchmark programs, including fast Fourier transform (FFT), the Dijkstra's shortest path algorithm (DJK), quick-sort (QS), and basic mathematics suit (BMS). Each trace is divided into fixed length sequences of instructions referred to as slices. The sizes of the slices examined are 1000, 2000, 5000, and 10000 instructions. Following a feature extraction stage applied to slices, each slice is represented by a set of architecture and micro-architecture features such as the number of integer operations per slice, the number of register transfers, and the number of memory accesses. These sets of features are going through the feature selection stage described below, where an optimal subset of the features is sought. Based on findings from the first phase of this research (which did not include caching), and due to external constraints, the 12 "best" features out of 24 features extracted per slice are selected for FFT and BMS and the best 13 out of 25 extracted features are selected for the DJK and QS. , Next, ISODATA¹ clustering [25], using the selected feature subset, identifies prototype slices, and classifies slices according to the closest prototype [1-2].

¹ The ISODATA clustering algorithm is an extension of the K-means clustering algorithm where the initial number of clusters can be changed through merge, split, and cluster elimination operations [25].

4.2 Non Cached GA in the FSP Domain

In the current implementation of the FSP, chromosomes represent feature subsets; encoded by bit strings with a length equal to the number of features. A 1-bit in a chromosome denotes that the respective feature is selected, and a 0-bit denotes a feature which is not selected. Mutations are implemented by applying a complement operation to randomly selected chromosome bits. Crossovers are implemented using the single-point crossover method [3]. Overall, the mutation operation is applied in 2% of the cases of chromosome generation and crossover in 98% of the cases. Since the number of desired features is given, (say n) and represented by the number of 1-bits in the chromosome, a valid chromosome must include n 1-bits. Nevertheless, since the mutations and crossovers might change the number of 1-bits in a chromosome, an operation of chromosome repair which randomly switches 0 or 1-bits, as required, follows the mutation and crossover operations.

The non cached version is running for up to 20,000 generations. A highest scoring selection rule (elitism) is implemented where the total population is 384 chromosomes and the elite list consists of the best 256 chromosomes observed in the population. Each consecutive generation is generating 128 additional chromosomes while maintaining the best 256 for the next generation.

Chromosome fitness is evaluated through ISODATA clustering. The features represented by the chromosome are used to cluster the training data. The resultant ratio of the “between” and “within” cluster dispersion matrices [25] is referred to as the quality (fitness) of the chromosome. Since ISODATA have several tunable parameters, a process of parameter tuning using principles of design of experiments has been performed prior to using the ISODATA procedure. Note that the ISODATA is a relatively complex algorithm which might require long execution time. Hence, it is desirable that the feature selection utility performs the minimal number of chromosome evaluations (or ISODATA calls).

4.3 Caching in the GA FSP Domain

The implementation of the GA with cache is done in the following way:

- 1) The first generation is executed in similarity to the non cache version. However, at the end of the execution of the first generation, the best 256 chromosomes are maintained in an elite list while the other 128 chromosomes are cached.
- 2) Next, a member of the elite list might be randomly selected for mutation. Alternatively, two elite members might be randomly selected for crossover, and randomly one of the two siblings is selected for further evaluation.
- 3) The new chromosome generated as the result of mutation or crossover is first compared (structurally) to the members of the elite list. If an identical chromosome resides in the elite list then the current chromosome is discarded and a new chromosome is generated through mutation or crossover of elite members.
- 4) Next, if the chromosome does not exist in the elite list then, it is compared (structurally) to the members of the cache. The comparison might result in a hit or a miss (see step 5). A hit means that this chromosome has already been evaluated; hence it is inferior to any member of the elite list, and can be discarded. Nevertheless, cache

counters are updated to reflect the hit as it might affect the replacement (eviction) decision in subsequent misses.

- 5) A miss is interpreted as an indication of distinct chromosome. It is recorded and miss counters are updated.
 - a) The chromosome fitness is evaluated, and compared to the elite list. If it is better than the fitness of the worst member of the elite list, then the new chromosome is inserted into the elite list and the worst chromosome of the elite list is cached.
 - b) Otherwise the new chromosome is inferior to the elite list. It is inserted to the cache.

4.4 Experiments Results

As mentioned in Section 4.1, the experiment instances include four benchmarks and the related code is segmented into different slice sizes. In addition, a large combination of cache configurations (number of sets and associativity) as well as replacement policies are used in each experiment. The GA with cache and no cache is run four times, on each experimental condition (defined by a particular benchmark, slice size, cache configuration, and replacement policy). The best clustering quality, the miss ratio, and the computational time are recorded at three different stopping points and averaged over the four replicates. The first stopping point results from running all GAs until a predetermined number of **distinct** chromosomes are reached. Then, GAs is running until a **pre-determined number of generations** is reached. The final stopping point occurs when a **pre-determined number of ISODATA clustering operations** is reached. In addition, this study calculates the inherent redundancy; as well as an estimation of the speedup of the GA due to caching in both software and hardware caching scenarios.

This research focuses on the hardware cache scenario as it is most likely to be used in practical applications. For this scenario, the number of ISODATA iterations is utilized as a useful “atomic time unit” to compare the execution time of the algorithms under different parameters such as cache sizes and replacement policies. One ISODATA iteration takes the same amount of time in every scenario relevant to our experiments. Moreover, all of the other parts of the algorithm including cache access, search, and update are negligible compared to the average execution time of one ISODATA iteration. Hence, the number of ISODATA iterations can give a relatively accurate estimate of the realistic speedup obtained with different cache configurations.

Figure 1, is a box plot that shows the spread in the number of ISODATAs after four replications for each benchmark studied. The boxes under the zero label represent the no cache case. Figure 2 displays the speedup for the different cache entry sizes. The speedup is computed as the ratio of the average number of ISODATAs for the no cache case (labeled as 0) to the average number of ISODATAs for a particular cache entry-size. For all the cases included in the figures, the slice-size is 1000 instructions, and the set associativity is 8. Statistical analysis of variance (ANOVA) over data collected from previous experiments shows that there is no significant difference between the three replacement policies studied; consequently, all experimental reported in Figures 1 and 2 are run under the least frequently used (LFU)

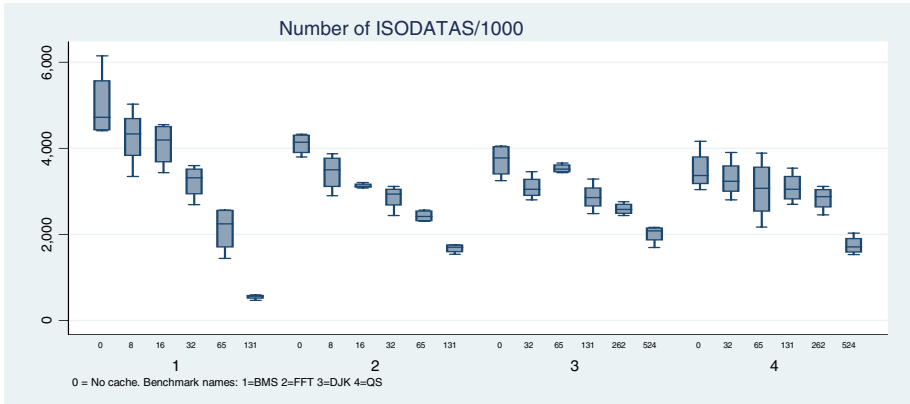


Fig. 1. Number of ISODATA calls per run

replacement policy. As figure 2 shows, a speedup of 8x is obtained with FFT while a speedup of about 2x is obtained with the rest of the benchmarks. The reason for the lower numbers in the other benchmarks is that their problem domain is larger and therefore a larger cache is required in order to exploit redundancy.

In addition, we measured the upper bound on the speedup attainable through the infinite memory model and obtained 50x for BMS, 41 for FFT, 37 for DJK, and 34 for QS. This shows that much more speedup can be obtained with larger cache sizes.

In addition to the box plots analysis, ANOVA analysis has been performed and yields the following conclusion with significance level of 95%: 1) The cache contribution is significant, 2) Cache replacement policies have the same contribution, 3) Cache sizes are significant, but associativity is not (hence we used only one associativity value in the graph), 4) Slice sizes are significant with best results obtained for a slice of 1000 instructions. 5) For a fixed number of generation there is no significant difference in the quality of the cached versus non-cached versions.

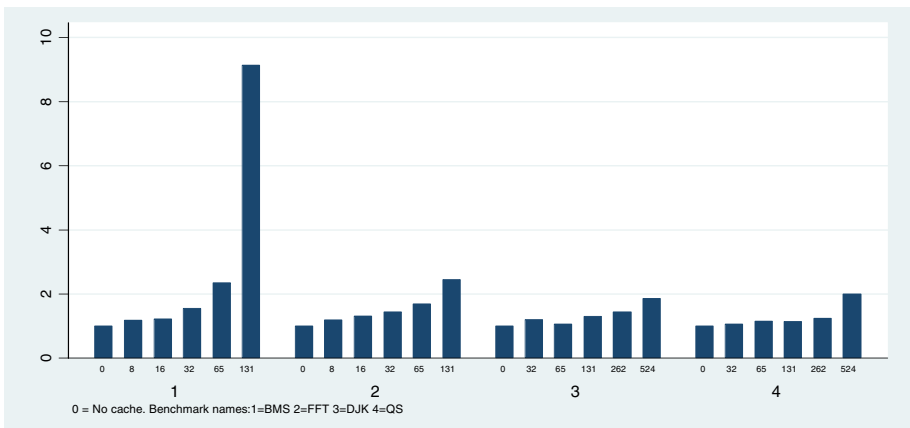


Fig. 2. Speedup results

Another finding of the research is that in general, if a time constraint is applied then the cache version is likely to provide a solution with about 2% better quality than the non-cached version. Nevertheless, eventually if the constrain is removed the non cached version achieves similar results.

5 Conclusions and Further Research

The experiments performed demonstrate that spatial and temporal locality of reference properties can be exploited in GA to reduce computational time without degrading solution quality. Overall, this research shows that adding a cache to the GA can result in significant speedup.

Several directions for future research can be considered including 1) exploring other record keeping mechanisms and other heuristics for solving the FSP such as ant colony, particle swarm, and tabu search, 2) extending the record keeping approach to other search techniques such as simulated annealing and other domains e.g., scheduling, and 3) Further analysis of the tradeoffs between parameters related to the search and storage overhead.

References

1. Luo, Y., Joshi, A., Phansalkar, A., John, L.K., Ghosh, J.: Analyzing and Improving Clustering Based Sampling for Microprocessors. *Journal of High Performance Computing and Networking* 5(4), 352–366 (2008)
2. Brock, M.: Feature Selection for Slice Based Workload Characterization and Power Estimation, Texas State University, Master Thesis (in progress)
3. Vose, M.D.: *The simple genetic algorithm: Foundations and theory*. MIT Press, Cambridge (1999)
4. Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters* 10, 335–347 (1989)
5. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition* (33), 25–41 (2000)
6. Kudo, M., Somol, P., Pudil, P., Shimbo, M., Sklansky, J.: Comparison of Classifier Specific Feature Selection Algorithms. In: Amin, A., Pudil, P., Ferri, F., Iñesta, J.M. (eds.) *SPR 2000 and SSPR 2000*. LNCS, vol. 1876, pp. 677–686. Springer, Heidelberg (2000)
7. Cover, T.M., Van Campenhout, J.M.: On the possible orderings in the measurement selection problem. *IEEE Transactions on Systems, Man and Cybernetics* 7(9), 657–661 (1997)
8. Guyon, I.J., Weston, S., Barnhill, V.: Gene selection for cancer classification using support vector machines. *Machine Learning* 46(1) (2002)
9. Gadat, S., Younes, L.: A stochastic algorithm for feature selection in pattern recognition. *Journal of Machine Learning Research* 8, 509–547 (2007)
10. Wang, Y.L., Li, J., Ni, S., Huang, T.: Feature selection using tabu search with long-term memories and probabilistic networks. *Pattern Recognition Letters* 30, 661–670 (2009)
11. Liu, H.: *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic, Boston (1998)
12. Bello, R.A., Puris, A., Nowe, Y., Martinez, G.M.: Two Step Ant Colony System to Solve the Feature Selection Problem. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) *CIARP 2006*. LNCS, vol. 4225, pp. 588–596. Springer, Heidelberg (2006)

13. Coetzee, F.M., Glover, E., Lawrence, S., Lee, C.: Feature selection in web applications using ROC insertions and power set pruning. In: Proceedings 2001 Symposium on Applications and the Internet, USA, pp. 5–14 (2001)
14. Narendra, P.M., Fukunaga, K.: A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers* 26, 917–922 (1977)
15. Nakariyakul, S.: On the suboptimal solutions using the adaptive branch and bound algorithm for feature selection. In: Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, pp. 384–389 (2008)
16. Yusta, S.C.: Different meta-heuristic strategies to solve the feature selection problem. *Pattern Recognition Letters* 30, 525–534 (2009)
17. Wang, X.J., Yang, X., Teng, W., Xia, R.: Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters* 28, 459–471 (2007)
18. Stone, H.S.: High-performance computer architecture. Prentice-Hall, Englewood Cliffs (1993)
19. Hertel, P., Pitassi, T.: An exponential time/space speedup for resolution. *Electronic Colloquium on Computational Complexity* 46, 1–25 (2007)
20. Allen, D., Darwiche, A.: Optimal time-space tradeoff in probabilistic inference. In: Proceedings of the 21st international joint conference on artificial intelligence, pp. 969–975 (2003)
21. Aggarwal, A.: Software caching vs. pre-fetching. In: ACM Proceedings of the International Symposium on Memory Management, Germany, pp. 1–6 (2002)
22. Karhi, D., Tamir, D.E.: Caching in the TSP Search Space. In: Chien, B.-C., Hong, T.-P., Chen, S.-M., Ali, M. (eds.) IEA/AIE 2009. LNCS, vol. 5579, pp. 221–230. Springer, Heidelberg (2009)
23. Santos, E.E., Santos Jr., E.: Cache Diversity in genetic algorithm Design. In: FLAIRS Conference, pp. 107–111 (2000)
24. Lowell, D., El Lababedi, B., Novoa, C., Tamir, D.E.: The Locality of Reference of Genetic algorithms and Probabilistic Reasoning. In: The International Conference on Artificial Intelligence and Pattern Recognition, AIPR 2009, USA (2009)
25. Ball, J.H., Hall, D.J.: A clustering technique for summarizing multivariate data. *Behavioral Science* 12(2), 153–155 (1966)

Learning Improved Feature Rankings through Decremental Input Pruning for Support Vector Based Drug Activity Prediction*

Wladimiro Díaz-Villanueva, Francesc J. Ferri, and Vicente Cerverón

Departament d'Informàtica, Universitat de València,
Av. Vicent Andrés-Estellés, s/n, 46100 Burjassot (València), Spain
{wladimiro.diaz,francesc.ferri,vicente.cerveron}@uv.es
<http://www.uv.es>

Abstract. The use of certain machine learning and pattern recognition tools for automated pharmacological drug design has been recently introduced. Different families of learning algorithms and Support Vector Machines in particular have been applied to the task of associating observed chemical properties and pharmacological activities to certain kinds of representations of the candidate compounds. The purpose of this work, is to select an appropriate feature ordering from a large set of molecular descriptors usually used in the domain of Drug Activity Characterization. To this end, a new input pruning method is introduced and assessed with respect to commonly used feature ranking algorithms.

Keywords: Support Vector Machines, Feature Selection, Feature Ranking, Pattern Classification, Pharmacological drug selection.

1 Introduction

Computer assisted pharmacological drug design is nowadays a challenging and very important problem in the pharmaceutical industry. The traditional approach for formulating new compounds requires the designer to test in the lab a very large number of molecular compounds, to select them in a blind way, and to look for the desired pharmacological property. Therefore, it is very useful to have tools to predict a priori the pharmacological activity of a given molecular compound in order to minimize laboratory experiments.

All methods developed for this purpose are based on the fact that the activity of a molecule derives from its chemical structure and therefore it is possible to find a relationship between this structure and the properties that the molecule exhibits [14]. Thus, the particular way the molecular structure is represented has special relevance.

* This work has been partially funded by Feder and Spanish MEC projects TIN2009-14205-C04-03 and Consolider Ingenio 2010 CSD2007-00018.

In Chemical Graph Theory, molecular structures are represented as doubly labeled graphs which can be conveniently characterized by a number of specific graph descriptors. In this context, a great number of topological indices have been proposed, but only a small subset is widely used in common practical studies. In this work, three different sets or families of topological indices are considered together. A set of 62 topological-structural indices [13], the well-known Kier-Hall indices [9] and the so called charge or electro-topological indices [6].

These or similar representations have already been applied to different discrimination problems in drug design (analgesic, antidiabetic, antibacterial, etc.). In the particular case of antibacterial activity, very good classification results have been reported using multilayer perceptrons (MLP) [12] and Support Vector Machines [5].

2 Drug Activity Prediction Using Machine Learning

The quantitative structure-activity relationship (QSAR) paradigm is currently used in the computer aided design of new medical drugs with desired chemical properties. These methods are an alternative to the exact or precise description of the electronic properties of a molecule calculated by mechanical-quantum methods. The molecular topology describes the molecule as a set of indices which are in fact graph invariants. These topological indices are numerical descriptors that encode information about the number of atoms and their structural environment. This representation is derived from the hydrogen-suppressed molecular formula seen as a graph [11,14].

The molecular topology considers a molecule as a planar graph where atoms are represented by vertices and chemical bonds are represented by edges. The chosen set of molecular descriptors should adequately capture the phenomena underlying the properties of the compound. In this work, a set of 116 indices has been selected from the three families considered [4] that we will be referred to as topological (62), Kier-Hall (22) and electro-topological (32).

These molecular representations have shown their ability for discriminating and predicting different kinds of pharmacological properties. Nevertheless, it is known that certain indices are more important than others for detecting particular cases. Obviously, the QSAR studies rely on the key fact that the activity of a molecule directly derives from its structure or, more precisely, from certain aspects of it. The better the chosen set of indices captures these particular aspects, the better the (blind) machine learning methods will characterize the activity of the molecule. As the molecular descriptors or indices have to be general in order to be applied in a wide range of drug design contexts, the ability of the particular learning methods used to capture non linear relations and high order dependencies among them becomes a key fact in the whole process. Moreover, the particular features that are most important for particular tasks and the order they need to be taken into account to obtain better results can lead to important information for characterizing future activity in the corresponding chemical compounds.

3 Ranking Inputs and Feature Selection

The selection of relevant features, and the elimination of irrelevant ones, is a central problem in machine learning [10]. Before an induction algorithm can move beyond the training data to make predictions about novel test cases, it must be decided which attributes are to be taken into account for these predictions and which ones must be ignored. Intuitively, one would like the learner to use only those attributes that are relevant to the target concept.

Selecting the optimal subset of features for a given learning or classification task is a well-known example of NP-hard problem [7]. For this reason, many different suboptimal algorithms that look for relevant attributes have been proposed. Among these, the family of sequential greedy subset selection algorithms is particularly appealing due to its very convenient trade-off between performance and computational demands.

In some cases, the particular optimal subset of features is not so important and a family of solutions of a convenient range of cardinalities is sought. More specifically and depending on the application domain, an absolute measure of relevance for each feature or a convenient ordering of them can be more interesting than particular families of subsets. This is indeed the case in Drug Activity Characterization when using descriptors coming from different families that are very different from the point of view of its origin and (chemical) definition.

4 Sensitivity Based Input Pruning

When the goal consists in obtaining a good feature ordering or ranking one can use classical forward or backward sequential selection (SFS and SBS algorithms) [7] that proceed by sequentially adding or discarding features in a greedy way. These algorithms are very efficient but they still require a quadratic number of model training and evaluation when used as wrappers [10] in order to obtain the corresponding feature ranking.

The so-called Sensitivity Based Pruning (SBP) [15,11] has been proposed for these kind of situations in the context of neural network models in which training a quadratic number of models can become prohibitive. Instead, the SPB performs a unique model training and a linear number of partial evaluations in order to compute a sensitivity measure for each feature. The sensitivity measure is computed as follows.

Let $f(x)$ be the function/classifier and let $x = (x_1, \dots, x_d)$ be its corresponding input. Let P_f be a particular performance measure on f using a given training set. The sensitivity of the i th feature/input is then given by

$$S_i = P_f - P_{f_i}$$

where $f_i(x) = f(x_1, \dots, \bar{x}_i, \dots, x_d)$ and \bar{x}_i is the average value of the i th input on the training set.

4.1 The Iterative Sensitivity Based Pruning (iSBP) Algorithm

In this work, we propose to use the SBP method in combination with a modified version of backward elimination. This method uses the ranking of features given by the SBP method and removes the least relevant variables one (or more) at a time. The resulting subset is then used to retrain the classifier and calculate a new ranking over the remaining features. This process is repeated until all features are removed. The details of the proposed procedure are shown as Algorithm 1.

The iSBP algorithm can be seen as a simplified SBS algorithm in which the SBS ranking is used instead of the tentative removal of all remaining features. In this way, the training of (about) $\frac{n^2}{2}$ models in sequential algorithms is substituted by $\frac{n}{m}$ models in the proposed procedure. The parameter m is used to improve even further the computational requirements of the algorithm. In this work, only results with $m = 1$ are considered and reported. Informal experimentation increasing m lead to very similar performance results with progressively less computational burden.

| |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Input: I training instances with J input features</p> <p>Input: m features to remove at each step</p> <p>begin</p> <p> while ($J > 0$) do</p> <p> Train a classifier with J input features;</p> <p> Ranking the input features using the SBP algorithm;</p> <p> Remove the m least significant features;</p> <p> $J \leftarrow J - m$;</p> <p> end</p> <p>end</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Algorithm 1. The proposed iSBP algorithm

5 Empirical Results

The particular discrimination problem considered in the present study was to determine whether a molecule has analgesic properties or not. To this end, a particularly difficult database has been compiled. A dataset of 973 samples with potential pharmacological activity has been considered. Out of these, 111 molecules are known to have analgesic properties while the other 862 compounds do not have these properties at all. These unbalanced proportions are not necessarily related to the a priori probability of activity in real pharmacological design trials. It is worth noting that one of the main drawbacks to obtain good and stable results with some of the methods considered in this work, comes from the small representativeness of the active class in the data set available. We will refer here to active and inactive compounds, respectively.

Support vector machines (SVM) have been used in this work as the learning algorithm given that this was the best option when studying a slightly different

discrimination problem involving drugs with or without antibacterial activity [5]. SVMs are very well-known, flexible and robust learning models that allow for very good classification and generalization [8,3]. In particular, the radial basis function (RBF) or proximity kernel parametrized by an influence parameter, γ , has been considered for the experiments. This parameter along with the soft margin or regularization parameter C have been chosen by looking at values similar to the ones in previous related works.

All 116 molecular descriptors were used to obtain feature vectors in which values were linearly normalized to the interval $[-1, 1]$ in an independent way. Each feature vector was then labeled either with 1 indicating that the molecule is active, i.e. it has analgesic properties, or -1 if the molecule is inactive.

The measure of performance used in this context both to derive the feature relevance and to assess the final classification results is the area under the ROC curve (AUC). The ROC curve conveniently measures the dependence of the active accuracy rate with regard to the inactive error rate (true positives against false positives) over the whole output range of the learned model. The AUC is a commonly used measure of the global accuracy of a classifier [2] that has been previously used for Automatic Drug Characterization because the particular cost-benefit tradeoff [5]. It is possible to define the relevance of a feature as the difference in the value of the AUC when the feature is removed. In this work, the AUC is estimated by numerical integration of the averaged ROC curve built from a n -fold cross validation.

In order to obtain results as significant as possible, n -fold stratified cross-validation (CV) has been used to compute all global accuracies and performance measures shown. In these cross-validation procedures, $n = 5$ has been used for the internal process of feature relevance estimation while $n = 10$ has been fixed in order to compute the final accuracy of the final models using the learned ordered list of features.

In order to make use of the available data as much as possible the experiments on feature ranking and assessment have been done using all the data. In other words, the whole dataset has been used to produce an ideally ordered list of features using all competing approaches considered. Once these lists have been obtained, each corresponding feature subset has been evaluated through a 10-fold cross-validation round using all available data again. In this way, the feature lists are the same across different folds and the results are slightly more stable and optimistically biased (but in an absolutely equal way for all competing feature ranking algorithms).

Figure 1 shows the accuracy results for the feature subsets obtained from the ordered lists computed by the four methods considered.

Apart from the SBS that gives the best results, one can observe that the proposed iSBS method gives better results than both SBP and SFS but for a relatively narrow margin. More important than that is the fact that iSBP is able to give very stable performance results in the cardinality range 8–40 which is of specific interest for this application domain given the observed intrinsic complexity of the classification problem.

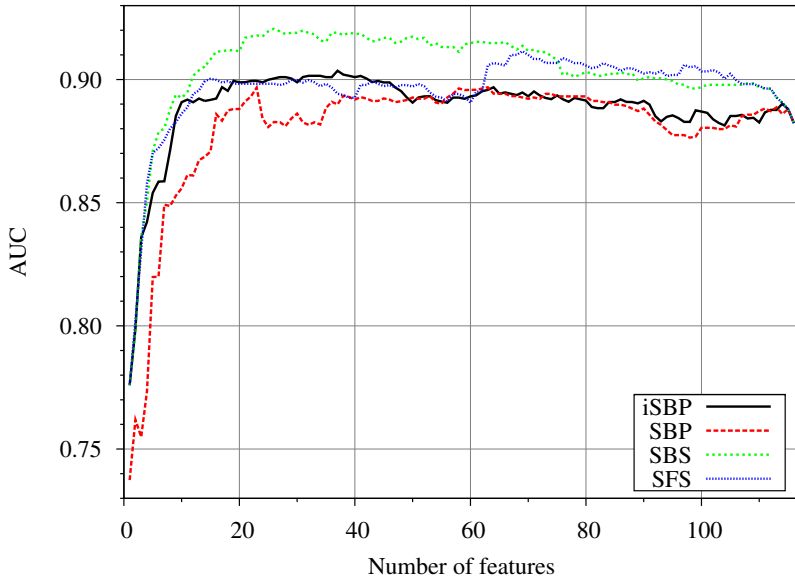


Fig. 1. Averaged AUC results obtained by the four considered algorithms in terms of the corresponding feature subsets derived from the rankings. For each method, a unique ranking is obtained.

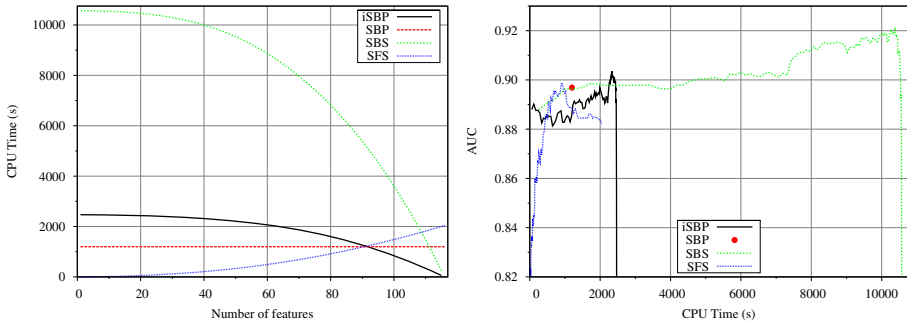


Fig. 2. Computational burden associated to each of the methods considered measured in CPU seconds used in the whole process. (left) CPU time against feature subset cardinality, (right) AUC rate against CPU time.

The computational cost associated to each method in terms of the corresponding feature subset size is shown in Figure 2 left. It can be seen that the best solutions given by SBS in the target range imply a computational burden that is almost five times higher. At the same time, the CPU time needed by iSBP is only twice the time of the basic SBS. The same CPU timing data is shown in Figure 2 right against corresponding accuracy values. This figure illustrates how iSBP gets to its maximum (using 37 features) much faster than

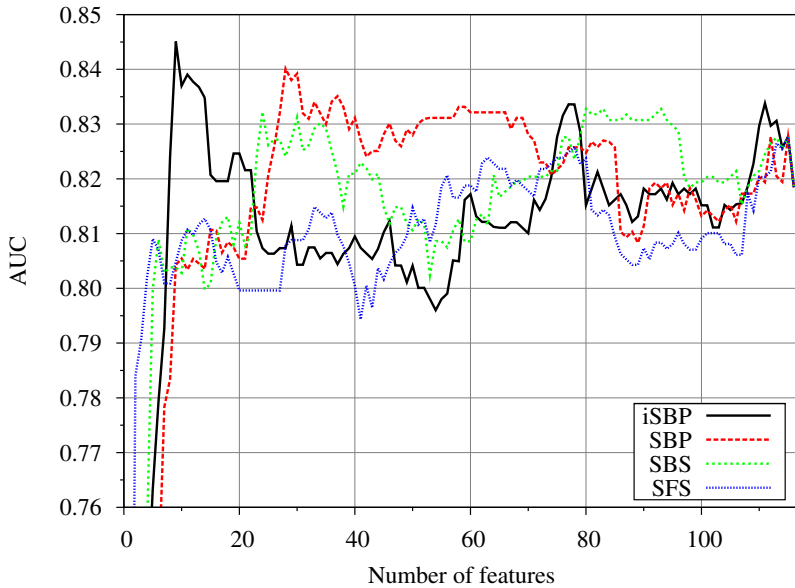


Fig. 3. Averaged AUC results obtained by the four considered algorithms in terms of the corresponding feature subsets derived from the rankings. For each method, different rankings are obtained and evaluated using different partitions of the available data.

SBS. Moreover, this plot also puts forward the very good ability of SFS to very quickly obtain good solutions. Unfortunately, the corresponding subset consists of about 80 features and no convenient solutions are found by SFS in the target range for this problem.

The same experimentation has been repeated in a more independent way by computing different feature lists for each different cross validation round in the assessment phase. In this way and as can be observed in Figure 3, averaged results are worse for all methods in absolute terms but relative considerations about the merits of different methods remain the same with a few exceptions. On one hand and as it could be expected, all results exhibit more variability. On the other hand, the SBS is no longer the best option.

Significantly better solutions both in terms of subset cardinality and accuracy are obtained with the proposed iSBP method even below 10 features. Competing results are obtained both with SBP and SBS in the range from 20 to 30 features while the SFS clearly gives the worst results for this particular and more demanding experiment.

6 Concluding Remarks and Further Work

An improved backward selection method aimed at obtaining convenient ordered lists of relevant features has been proposed and partially evaluated in a particular and challenging application domain. The proposed method has been compared

to the classical forward and backward feature selection and also to the sensitivity based input pruning method. All results are in terms of AUC measures computed from the outputs of conveniently learned SVM models using the corresponding subsets of relevant features of different cardinalities. The empirical results obtained show that only the three backward methods are able to obtain good results in the cardinality range which is interesting for this particular application domain, i.e. about one third of the original dimensionality. Moreover, a general conclusion is that the dramatic increase in computational burden needed by the SBS does not compensate for the improvement in performance. In fact, when measured more precisely, the results by SBS are even worse than the ones from the very simple SBP method. On the other, hand, the proposed method which can be seen as a convenient trade off between the other two backward options, keeps obtaining good solutions both in terms of accuracy and feature subset cardinality.

Further work is already being carried out in the direction of deeply studying to which extent the proposed algorithm is better than the other ones in a wider range of situations. The other line of research is related to the in depth study of the different features selected or ranked by each method. This could effectively lead to new proposals of molecular descriptors adapted to particular discriminating tasks.

References

1. Basak, S., Bertelsen, S., Grunwald, G.: Application of graph theoretical parameters in quantifying molecular similarity and structure-activity studies. *J. Chem. Inf. Comput. Sci.* 34, 270–276 (1994)
2. Dodd, L., Pepe, M.: Partial auc estimation and regression. *Biometrics* 59(3), 614–623 (2003)
3. Wang, L.P.: *Support Vector Machines: Theory and Application*. Springer, Berlin (2005)
4. Ferri, F.J., Díaz-Villanueva, W., Castro, M.J.: A comparative study using different topological representations in pattern recognition based drug activity characterization. In: *IEEE International Conference on Systems, Man and Cybernetics*, pp. 2011–2015 (2007)
5. Ferri, F.J., Diaz-Villanueva, W., Castro, M.: Experiments on automatic drug activity characterization using support vector classification. In: *IASTED Intl. Conf. on Computational Intelligence (CI 2006)*, San Francisco, US, pp. 332–337 (November 2006)
6. Galvez, J., Garcia, R., Salabert, M., Soler, R.: Charge indexes. new topological descriptor. *J. Chem. Inf. and Comp. Sciences* 34, 502–525 (1994)
7. Jain, A., Zongker, D.: Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Mach. Intell.* 19(2), 153–158 (1997)
8. Kecman, V.: *Learning and Soft Computing, Support Vector machines, Neural Networks and Fuzzy Logic Models*. The MIT Press, Cambridge (2001)
9. Kier, L.B., Hall, L.H.: *Molecular Connectivity in Structure-Activity Analysis*. John Willey and Sons, New York (1986)

10. Langley, P.: Selection of relevant features in machine learning. In: Proceedings of the AAAI Fall symposium on relevance, pp. 140–144. AAAI Press, Menlo Park (1994)
11. Moody, J., Utans, J.: Principled architecture selection for neural networks: Application to corporate bond rating prediction. In: Moody, J.E., Hanson, S.J., Lippmann, R.P. (eds.) *Advances in Neural Information Processing Systems*, vol. 4, pp. 683–690. Morgan Kauffmann Publishers, San Francisco (1992)
12. Murcia-Soler, M., Pérez-Giménez, F., García-March, F., Salabert-Salvador, M., Díaz-Villanueva, W., Castro-Bleda, M., Villanueva-Pareja, A.: Artificial neural networks and linear discriminant analysis: A valuable combination in the selection of new antibacterial compounds. *J. Chem. Inf. and Comp. Sciences* 3, 1031–1041 (2004)
13. Murcia-Soler, M., Pérez-Giménez, F., García-March, F., Salabert-Salvador, M., Díaz-Villanueva, W., Medina-Casamayor, P.: Discrimination and selection of new potential antibacterial compounds using simple topological descriptors. *J. Mol. Graph. Model* 21, 375–390 (2003)
14. Seybold, P., May, M., Bagal, U.: Molecular structure-property relationships. *J. Chem. Educ.* 64, 575–581 (1987)
15. Utans, J., Moody, J.E.: Selecting neural network architectures via the prediction risk: Application to corporate bond rating prediction. In: Proceedings of the First International Conference on Artificial Intelligence Applications on Wall Street, pp. 35–41. IEEE Computer Society Press, Los Alamitos (1991)

Scaling Up Feature Selection by Means of Democratization*

Aida de Haro-García and Nicolás García-Pedrajas

Department of Computing and Numerical Analysis
University of Córdoba, Spain
{adeharo, npedrajas}@uco.es

Abstract. The overwhelming amount of data that is available nowadays makes many of the existing machine learning algorithms inapplicable to many real-world problems. Two approaches have been used to deal with this problem: scaling up data mining algorithms [1] and data reduction. Nevertheless, scaling up a certain algorithm is not always feasible. One of the most common methods for data reduction is feature selection, but when we face large problems, the scalability becomes an issue. This paper presents a way of removing this difficulty using several rounds of feature selection on subsets of the original dataset, combined using a voting scheme. The performance is very good in terms of testing error and storage reduction, while the execution time of the process is decreased very significantly. The method is especially efficient when we use feature selection algorithms that are of a high computational cost. An extensive comparison in 27 datasets of medium and large sizes from the UCI Machine Learning Repository and using different classifiers shows the usefulness of our method.

Keywords: Feature Selection, Instance-based Learning, Classification, Huge Problems, Scale-up, Divide & Conquer, Data Mining, Dimensionality Reduction, Pattern Discovery, Ensembles, Genetic Algorithms, Relief Algorithms.

1 Introduction

Data mining [2], as a multidisciplinary joint effort from databases, machine learning, and statistics, is championing in turning mountains of data into nuggets. In order to use data mining tools effectively, data preprocessing is essential. Feature selection is one of the most important and frequently used techniques in data preprocessing for data mining [3, 4]. In contrast to other dimensionality reduction techniques, they preserve the original semantics of the variables, hence, offering the advantage of interpretability by a domain expert [5].

Feature selection can be defined as the selection of a subset of M features from a set of N features, $M < N$, such that the value of a criterion function is optimized over all subsets of size M [6]. However, the advantages of feature selection techniques come at a certain price, as the search for a subset of relevant features introduces an

* This work has been financed in part by the project “Constructing ensembles of classifiers by means of a new approach for boosting. Application to Bioinformatics recognition problems.” (TIN2008-03151) of the Spanish Ministry of Science.

additional layer of complexity in the modeling task. In this paper we propose a methodology to reduce this cost without severely damaging the classification accuracy.

A typical feature selection process consists of four basic steps (shown in Figure 1), namely, subset generation, subset evaluation, stopping criterion, and result validation [7].

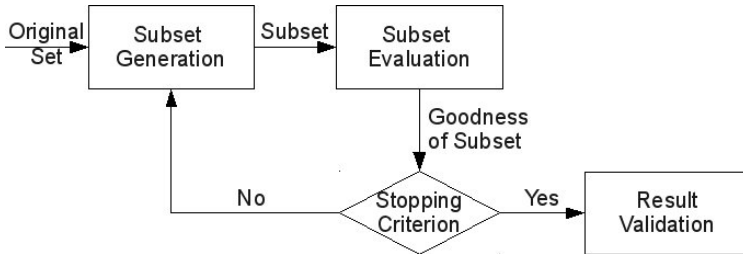


Fig. 1. Four basic steps of a typical feature selection process

The generation procedure is a search procedure [8] that produces candidate feature subsets for evaluation based on a certain search strategy. Feature selection methods applied in this paper generate candidates randomly.

An evaluation function measures the goodness of the subset produced and this value is compared with the previous best. If it is found to be better, then it replaces the previous best subset.

The process of subset generation and evaluation is repeated until a given stopping criterion is satisfied (e.g.: a predefined number of features are selected or a number of iterations reached). The feature selection process ends by outputting a selected subset of features to a validation procedure [7].

In the context of classification, feature selection techniques can be organized into three categories, depending on how they combine the feature selection search with the construction of the classification model [5]: filter methods [9, 10, 11], wrapper methods [12, 13, 14], and hybrid / embedded methods [15, 16, 17]:

- Filter techniques rely on the intrinsic properties of the data to evaluate and select feature subsets without involving any mining algorithm. They easily scale to very high-dimensional datasets, they are computationally simple and fast, and independent of the classification algorithm.
- Wrapper methods embed the model hypothesis search within the feature subset search. The evaluation of a specific subset of features is obtained by training and testing a specific classification model. Their advantages include the interaction between feature subset search and model selection, and the ability to take into account feature dependencies. A common drawback is that they have a high risk of over-fitting and are very computationally intensive.
- Hybrid / embedded techniques attempt to take advantage of the two models by building the search for an optimal subset of features into the classifier construction. Just like wrapper, they are specific to a given learning algorithm.

As previously stated, when the dimensionality of a domain expands the number of features N increases. In these cases, finding an optimal feature subset is usually intractable

[14] and many problems related to feature selection have been shown to be NP-hard [18]. In order to face this problem in this paper we propose a methodology for applying feature selection algorithms based on repeating several rounds of a fast feature selection process. Each round on its own would not be able to achieve a good performance. However, the combination of several rounds using a voting scheme is able to match the performance of a feature selection algorithm applied to the whole dataset with a large education in the time of the algorithm. Each round can be considered a weak feature selector, as it has a partial view of the dataset, their combination using a voting scheme is similar to the combination of different learners in an ensemble using a voting scheme. Due to this voting scheme we call this method *democratization* of feature selection algorithms, and the result *democratic feature selection*.

The main advantage of our method is that as the feature selection algorithm is applied only to small subsets, the time is reduced very significantly. In fact, as the size of the subset is chosen by the researcher, we can apply the method to any problem regardless of its size. As for the case of classifier ensembles, where the base learner is a parameter of the algorithm, in our method the feature selection method is a parameter, and any algorithm can be used.

This paper is organized as follows: Section 2 presents the proposed model for feature selection based on our approach; Section 3 reviews some related work; Section 4 describes the experimental setup; Section 5 shows the results of the experiments; and Section 6 states the conclusions of our work and future research lines.

2 Democratic Feature Selection Method

The process consists of dividing the original dataset into several disjoint subsets of approximately the same size that cover all the dataset. Then, the feature selection algorithm is applied to each subset separately. The features that are selected by the algorithm to be removed receive a vote. Then, a new partition is performed and another round of votes is carried out. After the predefined number of rounds is made, the features which have received a number of votes above a certain threshold are removed. An outline of the method is shown in Algorithm 1. Each round can be considered to be similar to a classifier in an ensemble, and the combination process by voting is similar to the combination of base learners in bagging or boosting [19].

Algorithm 1. Democratic feature selection (DemoFS) algorithm

Data: A training set $T = \{(x_1, y_1), \dots, (x_n, y_n)\}$, subset size s , and number of rounds r .

Result: The set of selected features

```

for i = 1 to r do
  Divide data into  $n_s$  subsets of size  $s$ , by instances or by features
  for j = 1 to  $n_s$  do
    Apply feature selection algorithm to  $t_j$ 
    Store votes of removed features from  $t_j$ 
  end
end
Obtain threshold of votes,  $v$ , to remove a feature
 $S = T$ 
Remove from  $S$  all features with a number of votes above or equal to  $v$ 
return  $S$ 

```

The most important advantage of our method is the large reduction in execution time. The reported experiments will show a large difference when using standard widely used feature selection algorithms. Additionally, the method is easy to implement in a parallel environment, as the execution of the feature selection algorithm over each subset is performed independently. Furthermore, as the size of the subsets is a parameter of the algorithm, we can choose the complexity of the execution in each one of the processors.

The partition method used in our approach is a strictly random partition. A proposed further work is to develop a data - dependent partition method, in order to improve the method efficiency.

However, as stated so far, the method still has an important issue to be addressed before we can obtain a useful algorithm: the determination of the number of votes, which is problem-dependent. Depending on the problem, a certain threshold may be too low or too high. Thus a method must be developed for the automatic determination of the number of votes needed to remove a feature from the training set. The automatic determination of this threshold has the additional advantage of relieving the researcher of the duty of setting a difficult parameter of the algorithm. This issue is discussed in the following section. We must also emphasize that our method is applicable to any feature selection algorithm, as the feature selection algorithm is a parameter of the method. Moreover our approach has a very competitive complexity, as it is linear in the number of instances or in the number of features (depending on which factor relies the complexity of the feature selection base algorithm).

2.1 Determining the Threshold of Votes

An important issue in our method is determining the number of votes needed to remove a feature from the training set. As it highly depends on the specific dataset, we need a way of selecting this value directly from the dataset in run time. Our choice is estimating the best value for the number of votes from the effect on the training set, specifically using a 10% of the dataset so as to speed up the process.

The election of the number of votes must take into account two different criteria: training error, e_r , and memory requirements (percentage of features retained), m . Both values must be minimized as much as possible. Our method of choosing the number of votes needed to remove a feature is based on obtaining the threshold number of votes, v , that minimizes a fitness criterion, $f(v)$:

$$f(v) = \alpha \cdot e_r(v) + (1 - \alpha) \cdot m(v). \quad (1)$$

α is a value in the interval $[0, 1]$ which measures the relative relevance of both values. In general, the minimization of the error is more important than storage reduction, thus, we have used a value of $\alpha = 0.75$.

We perform r rounds of the algorithm and store the number of votes received by each feature. Then, we must obtain the threshold number of votes, v , in the interval constituted by the minimum and maximum number of votes received by any feature. We calculate the criterion $f(v)$ (see Formula 1) for all the possible threshold values from 1 to r , and assign to v the value which minimizes the criterion. After that, we perform the feature selection removing the features whose number of votes is above or equal to the obtained threshold v .

3 Related Work

There are not many previous works that have dealt with scaling up feature selection problems. It is worth mentioning the work of Robnik-Sikonja [20]. They proposed a method to speed up the Relief algorithm by means of k-d trees. Although the algorithm shows a very good performance, k-d trees add new memory requirements and are less efficient if the trees are not balanced.

The idea of applying a recursive divide-and-conquer process to feature selection is inspired in a recent paper of the authors [21] that showed a good performance in the application of a democratic approach to instance selection, while attaining a dramatic reduction in the execution time of the instance selection process.

4 Experimental Setup

In order to make a fair comparison between the standard algorithms and our proposal, we have selected a set of 27 problems from the UCI Machine Learning Repository. For estimating the storage reduction and generalization error we used a 10-fold cross-validation (cv) method.

The evaluation of a certain feature selection algorithm is not a trivial task. We can distinguish two basic approaches: direct and indirect evaluation [22]. Direct evaluation evaluates a certain algorithm based exclusively on the data. The objective is to measure at which extent the selected features reflect the information present in the original data. Some proposed measures are entropy, moments, or histograms.

Indirect methods evaluate the effect of the feature selection algorithm on the task at hand. So, if we are interested in classification we evaluate the performance of the used classifier with the reduced set of inputs obtained after feature selection.

Therefore, when evaluating feature selection algorithms, the most usual way of evaluation is estimating the performance of the algorithms on a set of benchmark problems. In those problems several criteria can be considered, such as [23]: storage reduction, generalization accuracy, noise tolerance, and learning speed. Speed considerations are difficult to measure, as we are evaluating not only an algorithm but also a certain implementation. However, as the main aim of our work is scaling up feature selection algorithms, execution time is a basic issue. To allow a fair comparison, we have performed all the experiments in the same machine, a 33 blade chassis, each blade being a M600 Quad-Core Xeon E5420, with 2.5GHz and 2x6MB RAM memory. Our approach is based on applying feature selection algorithms to subsets of the training set, so to perform sound experiments the algorithm used for the whole training set and the algorithm used in our method are exactly the same. That is, when we applied our method using a feature selection algorithm and when we perform the feature selection algorithm for the whole training set, the implementation is the same in both cases.

The source code, in C and licensed under the GNU General Public License, used for all methods as well as the partitions of the datasets are freely available upon request to the authors.

4.1 Feature Selection Algorithms

In order to obtain an accurate view of the usefulness of our method, we must select some of the most widely used feature selection algorithms. We have chosen to test our model with two successful algorithms: ReliefF and a genetic algorithm.

ReliefF Algorithms. Relief family of algorithms is general, efficient and successful attribute estimators that do not assume the independence of the attributes and their quality estimates have a natural interpretation.

A key idea of the original Relief algorithm [24] is to estimate the quality of attributes according to how well their values distinguish between instances that are near to each other, features whose weights exceed a user-determined threshold are selected in designing the classifier. The complexity of Relief for a data set with n instances is $O(mkn)$, m being the user-defined iterations. The ReliefF algorithm [25] is not limited to two class problems, is more robust and can deal with incomplete and noisy data.

Genetic Algorithms. We apply a simple genetic algorithm. In a GA approach, a given feature subset is represented as a binary string (a “chromosome”) of length n , with a zero or one in position i denoting the absence or presence of feature i in the set. Note that n is the total number of available features. We apply standard genetic operators such as two-point crossover and mutation. At the beginning of each generation we perform an elitism step, and each individual is evaluated by means of the fitness function in Formula 2:

$$\text{fitness(ind)} = \text{suc_rate}(i) \cdot \alpha + (1 - \alpha) \cdot 1 - (n_sel_feat / n_feat), \quad (2)$$

where *suc_rate* is the wrapper evaluation of the subspace using classifier, α is a value in the interval $[0, 1]$ which is set to 0.75, n_sel_feat is the number of selected features and n_feat is the number of all available features.

5 Experimental Results

The same parameters were used for the standard version of every algorithm and its application within our methodology.

In the genetic algorithm the number of generations is set to 1000. We speed up the process applying an exhaustive search of all the possible combination of features if the exhaustive search has lesser iterations than the number of generations set in the genetic algorithm. At the beginning of each generation we apply a 10% of elitism and afterwards we get the rest of the population by means of applying iteratively a two-point-crossover operator (in which we keep the two best individuals of each crossover step). The standard mutation percentage is fixed to 10%.

In the ReliefF the number of iterations, m , is set to the number of available instances in the dataset, and the threshold to remove features is set to 0.05 (or 0.01 in those datasets which may find the previous value too restrictive). The number of neighbors, k , controls the locality of the estimates, and is set to 10 neighbors as recommended by its author (Kononenko, 1994).

Our method has three parameters: instance subset size or feature subset size, s , number of rounds, r , and α . We use an instance subset size of 100 individuals or 7 columns of feature subset size, as it is a value large enough to allow for a meaningful application of the feature selection algorithm on the subset, and small enough to allow for a fast execution. For the number of rounds we have chosen a small value to allow for a fast execution, $r = 10$. As explained in Section 2.1, α is set to 0.75, because for us it is more important a lesser error than little storage requirements.

5.1 Summary of Results

In both experiments, using ReliefF and the genetic algorithm, we employ a k-NN classifier to evaluate the subsets of features resulting in each round and later set the threshold to delete features.

To summarize these two experiments we show six figures that compare the results obtained from our methodology (dividing by instances and by features) and the classic approach in terms of testing error, storage requirements and execution time.

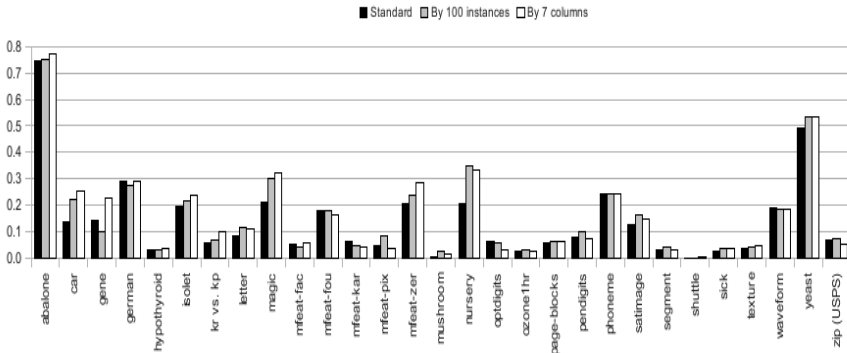


Fig. 2. Testing error for standard genetic algorithm and our approach dividing by instances and features

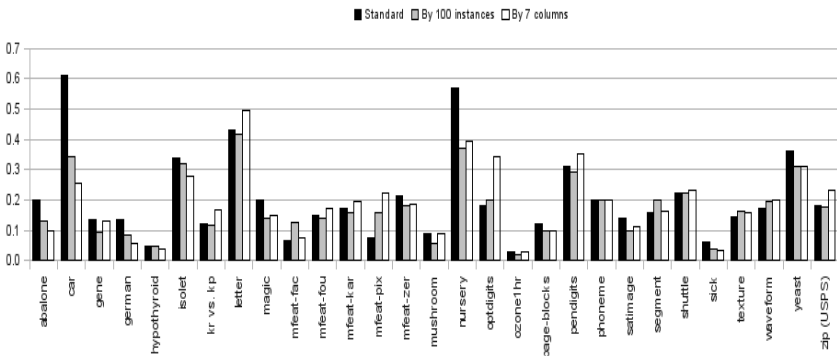


Fig. 3. Storage requirements for standard genetic algorithm and our approach dividing by instances and features

Results for the genetic algorithm are plotted in Figures 2, 3, 4. In terms of testing error our results dividing by instances are very similar to the standard ones. If we divide by features the results are a little bit worse, we believe that for this setting we need a bigger feature subset size (groups of 10 or 15 features) to be more reliable. Regarding the storage requirements we can observe very similar average results for all approaches. However, some datasets show worse results for our approach (e.g. mfeat-pix, opt-digits) and some of them are better (e.g. car or nursery in which the standard approach doesn't reduce much).

In terms of execution time, the advantage of our methodology is remarkable. For small problems there is a small overload due to the 10 rounds of votes performed, however as the problem grows in complexity, our approach shows a large reduction in the time needed to perform the feature selection process.

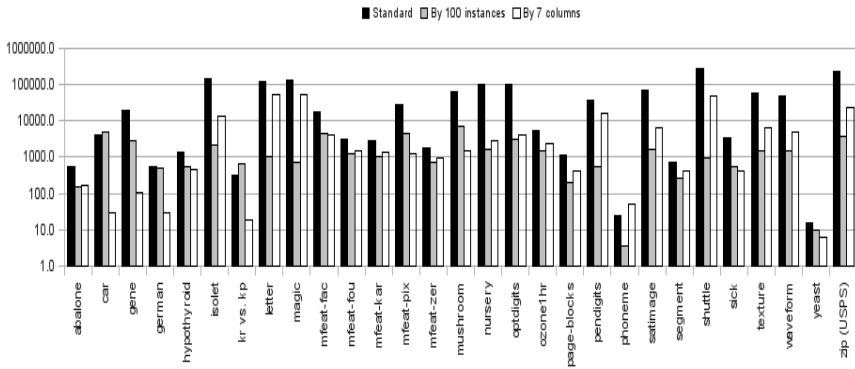


Fig. 4. Execution time for standard genetic algorithm and our approach dividing by instances and features

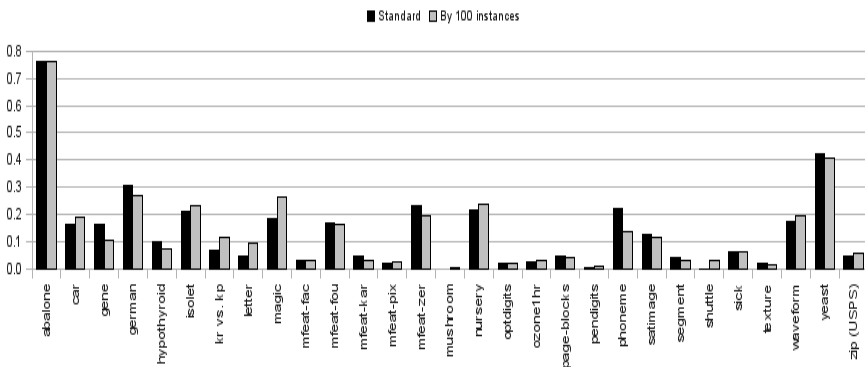


Fig. 5. Testing error for standard ReliefF algorithm and our approach dividing by instances and features

Results for the ReliefF algorithm are plotted in Figures 5, 6, 7. Regarding the testing error the standard method and our democratized counterpart have very similar performance in average. In terms of storage requirements we can observe a similar behavior in all approaches. Nevertheless, on the one hand, some datasets point out worse results for our approach (e.g. mfeat-fac, mfeat-pix, ozone1hr) but, on the other hand, some datasets perform better (e.g. gene, mushroom, sick). The running time is significantly reduced with our methodology, almost every dataset is improved.

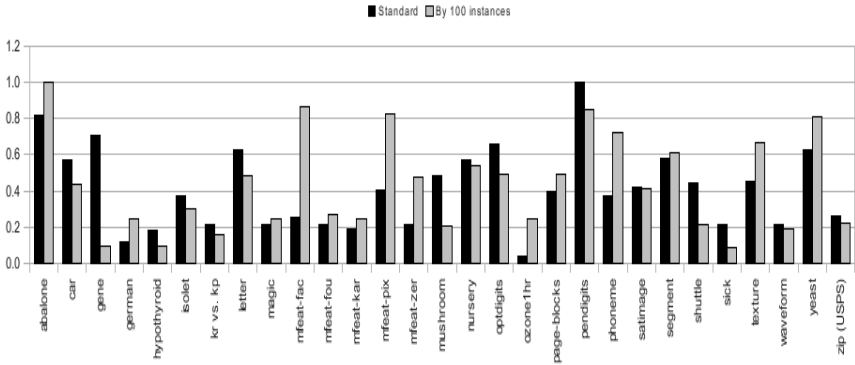


Fig. 6. Storage requirements for standard ReliefF algorithm and our approach dividing by instances

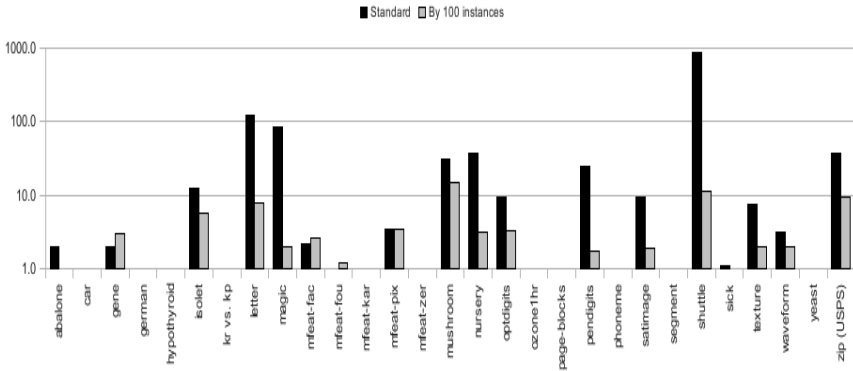


Fig. 7. Execution time for standard ReliefF algorithm and our approach dividing by instances

6 Conclusions and Future Work

In this paper we have presented a new method for scaling up feature selection algorithms. The method is applicable to any feature selection method without any modification. The method consists of performing several rounds of applying feature selection on disjoint subsets of instances or features of the original dataset and combining them by

means of a voting method. We use two well known feature selection algorithms: ReliefF and genetic algorithm. We have shown that our method is able to reduce considerably the running time, achieving a similar performance to the original algorithms. In terms of reduction of storage requirements and testing error, our approach is able to match and in some cases even improve the results obtained by means of the original feature selection algorithms over the full-size datasets. Additionally, our method is straightforwardly parallelizable without significant modifications.

As main research line we are working on the development of data – dependent methods to partition the original dataset. We expect that these partitions methods may have a positive influence on the performance of the method. Additionally we are working on a new hybrid approach with a filter step to select accurately the relevant features and a wrapper step to set efficiently a threshold of votes.

References

1. Provost, F.J., Kolluri, V.: A Survey of Methods for Scaling up Inductive Learning algorithms. *Data Mining and Knowledge Discovery* 3, 131–169 (1999)
2. Agrawal, R., Imielinski, T., Swami, A.: Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering* 5(6), 914–925 (1993)
3. Blum, A.L., Langley, P.: Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence* 97, 245–271 (1997)
4. Liu, H., Motoda, H.: Feature Extraction, Construction and Selection: A Mining Perspective. Kluwer Academic Publishers, Boston (2001)
5. Saeys, Y., Inza, I., Larranaga, P.: A Review of Feature Selection Techniques in Bioinformatics. *Bioinformatics* 23, 2507–2517 (2007)
6. Narendra, P.M., Fukunaga, K.: A Branch and Bound Algorithm for Feature Selection. *IEEE Transactions on Computers* C-26(9), 917–922 (1977)
7. Dash, M., Liu, H.: Feature Selection for Classification. *Intelligent Data Analysis* 1, 131–156 (1997)
8. Blum, A.L., Langley, P.: Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence* 97, 245–271 (1997)
9. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature Selection for Clustering - a Filter Solution. In: *Second International Conference on Data Mining*, pp. 115–122 (2002)
10. Liu, H., Setiono, R.: A Probabilistic Approach to Feature Selection - a Filter Solution. In: *Thirteenth International Conference on Machine Learning*, pp. 319–327 (1996)
11. Yu, L., Liu, H.: Feature Selection for High-dimensional Data: a Fast Correlation-based Filter Solution. In: *Twentieth International Conference on Machine Learning*, pp. 856–863 (2003)
12. Caruana, R., Freitag, D.: Greedy Attribute Selection. In: *Eleventh International Conference on Machine Learning*, pp. 28–36 (1994)
13. Dy, J.G., Brodley, C.E.: Feature Subset Selection and Order Identification for Unsupervised Learning. In: *Seventeenth International Conference on Machine Learning*, pp. 247–254 (2000)
14. Kohavi, R., John, G.H.: Wrappers for Feature Subset Selection. *Artificial Intelligence* 97, 273–324 (1997)
15. Das, S.: Filters, Wrappers, and a Boosting-based Hybrid for Feature Selection. In: *Eighth International Conference on Machine Learning*, pp. 74–81 (2001)

16. Das, S.: Filters, Wrappers, and a Boosting-based Hybrid for Feature Selection. In: Eighteenth International Conference on Machine Learning, pp. 74–81 (2001)
17. Xing, E.P., Jordan, M.I., Karp, R.M.: Feature Selection for High-dimensional Genomic Microarray Data. In: 18th International Conf. on Machine Learning, pp. 601–608. Morgan Kaufmann, San Francisco (2001)
18. Blum, A.L., Rivest, R.L.: Training a 3-node Neural Networks is NP-complete. *Neural Networks* 5, 117–127 (1992)
19. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the Margin: A new Explanation for the Effectiveness of Voting Methods. *The Annals of Statistics* 26, 1651–1686 (1998)
20. Robnik Sikonja, M.: Speeding up Relief Algorithm with K-d Trees. In: Electrotechnical and Computer Science Conference ERK 1998, Slovenia, pp. 137–140 (1998)
21. García-Osorio, C., de Haro-García, A., García-Pedrajas, N.: Democratic Instance Selection: a Linear Complexity Instance Selection Algorithm Based on Classifier Ensemble Concepts. *Artificial Intelligence* (accepted)
22. Liu, H., Motoda, H.: On Issues of Instance Selection. *Data Mining and Knowledge Discovery* 6, 115–130 (2002)
23. Wilson, D.R., Martinez, T.R.: Reduction Techniques for Instance-based Learning Algorithms. *Machine Learning* 38, 257–286 (2000)
24. Kira, K., Rendell, L.A.: A Practical Approach to Feature Selection. In: ML 1992 ninth international workshop on Machine learning, pp. 249–256. Morgan Kaufmann Publishers, San Francisco (1992)
25. Kononenko, I.: Estimating Attributes: Analysis and Extensions of Relief. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)

Author Index

- Abiyev, Rahib H. III-518
Abu Bakar, Norsharina III-359
Adeodato, Paulo J.L. II-357
Aguirre, Guillermo I-701
Ahn, Chang Wook II-126
Alaiz, Héctor I-264
Alaiz-Rodríguez, Rocío I-284
Alcalá, Rafael II-228
Alcalde, Cristina II-183
Alegre, Enrique I-284
Al-Fayoumi, Mohammad II-484
Alhanjouri, Mohammed I-388
Ali, Aida III-359
Al-Obeidat, Feras II-484
Alonso, Ángel I-264
Alonso del Rosario, Jose J. I-458
Alonso-González, Carlos J. II-96, II-116
Alonso-Nanclares, Lidia III-112
Álvarez-Aranega, P.J. I-294
Álvarez-García, Juan Antonio II-470
Alves Meira, Carlos Alberto II-337
Anacleto, Junia Coutinho I-215
Ansuategi, A. III-508
Aránega, A. I-294
Arangú, Marlene III-219
Arie, Hiroaki III-42
Aristondo, J. III-508
Arnaud, Adrian L. II-357
Arroyo, Ángel I-437
Arruda, L.V.R. III-546
Astarloa, Armando III-281
Aznarte M., José Luis II-239
- Badenas, Jorge I-368
Bader, Florian II-327
Badreddine, Ahmed II-595
Baena-García, Manuel I-560
Bahamonde, Antonio II-337
Bajo, Javier III-556
Balbi, Anahí I-123
Ballesteros-Yañez, Inmaculada III-129
Balthazar, José Manoel II-308
Banerjee, Soumya II-484
Baños, Orestí II-637
- Barber, Federico I-742, I-752, III-219
Barranco-López, Vicente I-489
Barranco, Manuel J. III-409
Barrenechea, Edurne III-369
Barrientos, Francisco II-193
Batet, Montserrat I-274
Bautista, J. III-656
Baykan, Nurdan Akhan II-47
Bayoumi, Abdel I-590
Becourt, Nicolas I-468
Beliakov, G. III-399
Bella, Antonio I-520
Ben Amor, Nahla II-595
Benavides, Carmen I-264
Benavides-Piccione, Ruth III-129
Ben Hamza, Abdessamad II-418
Benítez, José M. II-239
Bentahar, Jamal II-418
Beoldo, Andrea II-450
Berger, Nicolas I-82
Berg, Stian III-199
Berlanga, Rafael II-504
Bermejo, Pablo I-580
Bérubé, Hugo III-92
Bidarte, Unai III-281
Bielza, Concha I-531, III-149
Birattari, Mauro I-41
Blagojevic, Rachel I-358
Boissier, Olivier II-367
Borchani, Hanen I-531
Borrego-Jaraba, Francisco III-229
Borzemski, Leszek II-347
Bosse, Tibor II-407, II-565
Boulila, Zied III-139
Box, Braden II-494
Braga, Petrônio L. II-357
Breaban, Mihaela Elena II-67
Brighenti, Chiara III-21
Bugarín, Alberto I-175
Burillo, Mateo I-681
Burusco, Ana II-183
Bustince, Humberto III-341, III-369,
III-399

- Çakır, Hüseyin II-28
 Calvo, Óscar I-437
 Canada Bago, J. II-203
 Candela, Roberto II-288
 Canizes, Bruno I-731
 Cara, Ana Belén II-212
 Carbo, Javier III-470
 Carbone, Francesco II-18
 Cardou, Philippe I-82
 Carmona, Cristóbal I-601
 Carmona, Enrique J. III-169
 Carmona-Poyato, A. III-350
 Caro, Stéphane I-82
 Cases, Blanca III-538
 Castaño, Bonifacio I-72
 Castillo, José Carlos I-348
 Castillo, José M. III-389
 Cerruela García, Gonzalo I-458
 Cerverón, Vicente II-653
 Cesta, Amedeo I-154
 Chaquet, José M. III-169
 Chavarette, Fábio Roberto II-308
 Cheang, Brenda I-31
 Chen, Guan-Wei III-189
 Chen, Li I-235
 Chen, Pei-Yu III-189
 Chen, Wen-Pao III-271
 Chiang, Yi-Ting I-417
 Chica, M. III-656
 Choi, Hyukgeun I-651
 Choraś, Michał I-671
 Chu, Kuo-Chung III-189
 Ciardelli, Lorenzo II-450
 Ciferri, Cristina D.A. I-306
 Ciferri, Ricardo R. I-306
 Cleger-Tamayo, Sergio III-478
 Climent, Laura I-752
 Codina, Lluís III-488
 Cohn, A.G. III-321
 Contreras, Jesús II-18
 Corchado, Emilio III-636
 Corchado, Juan M. I-407, III-556
 Cordón, O. III-656
 Cornelis, Chris III-450
 Corral, Roque III-169
 Cottone, Giulio II-288
 Couto, Pedro III-341
 Crépin, Ludivine II-367
 Cristani, Matteo I-195
 Crowe, Malcolm II-615
 Cruz-Ramírez, M. III-646
 Curiel, Leticia III-636
 Cuxac, Pascal III-139
 Da Costa Pereira, Célia II-397, II-555
 Damarla, Thyagaraju II-605
 Damas, S. III-656
 D'Anjou, Alicia III-538
 da Silva, Marcus Vinícius Carvalho II-143
 De Asís, Agustín I-397
 Debenham, John I-113, II-377, II-387, III-72
 Deb, Kaushik III-379
 De Bock, Koen W. II-57
 de Castro, Juan Pablo I-225
 DeFelipe, Javier III-112, III-129
 de Haro-García, Aida II-662
 de la Cal, Enrique III-636
 Delsing, Jerker III-260
 De Maio, Carmen III-460
 Demazeau, Yves II-367
 Deneche, Abdel Hakim II-136
 Depaire, Benoît I-691
 De Paz, Juan F. III-556
 Derrac, Joaquín I-601
 de San Pedro, E. III-576
 de Souza, Luzia Vidal II-247
 Dias, Ana Luiza I-215
 Díaz-Villanueva, Wladimiro II-653
 Dragoni, Mauro II-555
 Drwal, Maciej II-347
 Errasti-Alcalá, Borja I-427
 Errecalde, Marcelo I-550, I-701
 Escot-Bocanegra, David I-427
 Etaati, Leila I-254
 Faghihi, Usef II-438
 Famili, Fazel III-92, III-102
 Faria, Luiz I-143
 Faria, Pedro I-731
 Fauteux, Francois III-102
 Featherston, Jonathan III-62
 Felfernig, Alexander I-621, I-631, I-641
 Fenza, Giuseppe III-460
 Fernández-Caballero, Antonio I-348
 Fernández de Alba, José M. I-448
 Fernandez-Garcia, N.L. III-350
 Fernandez, J. III-399
 Fernández, J.C. III-646

- Fernández, Jaime III-129
 Fernández, Laura III-129
 Fernández-Luna, Juan M. III-478
 Fernández-Navarro, F. III-1, III-646
 Fernández-Prieto, J.A. II-203
 Fernández-Recio, Raúl I-427
 Fernández-Robles, Laura I-284
 Ferrández, J.M. III-159
 Ferrarons, Pere III-21
 Ferreira, Rubem Euzébio II-164
 Ferretti, Edgardo I-701
 Ferri, Cèsar I-520
 Ferri, Francesc J. II-653
 Fileccia Scimemi, Giuseppe II-288
 Fiol-Roig, Gabriel I-185
 Flizikowski, Adam I-671
 Flores, M. Julia I-570
 Fobert, Pierre III-92
 Forcada, Francisco José I-368
 Fouriner-viger, Philippe II-438
 Franco, Leo I-317
 Frías, María Pilar I-205
 Fuentes-González, Ramón II-183
 Fujita, Hamido III-419
 Fu, Li-Chen I-417
 Fyfe, Colin I-327, II-257, II-615, II-627

 Gabrielli, Nicoletta I-195
 Gacto, María José II-228
 Gadeo-Martos, M.A. II-203
 Gaeta, Matteo III-460
 Galar, Mikel III-369
 Galichet, Sylvie I-468
 Gámez, José A. I-570, I-580
 Gámez, Juan Carlos III-301
 García-Fornes, Ana III-586
 García, Francisco II-12
 García-Gutiérrez, Jorge I-378
 Garcia, Isaías I-264
 García, Jesús III-498
 García Jiménez, Beatriz III-82
 García, Juan II-12
 García, Óscar I-407
 García-Osorio, César II-87, II-106
 García-Pedrajas, María D. I-327
 García-Pedrajas, Nicolás I-327, II-662
 García, Salvador I-601
 García-Torres, Miguel I-611
 García, Vicente I-541
 Garrido, Antonio I-244

 Garza-Castañón, Luis E. III-31
 Gerritsen, Charlotte II-565
 Ghribi, Maha III-139
 Gibert, Karina I-274
 Giordani, Stefano I-721
 Goldsztejn, Alexandre I-82
 Gómez-Luna, Juan III-389
 Gómez-Nieto, Miguel Ángel III-229
 Gómez, P. III-159
 Gómez-Pulido, Juan Antonio II-153,
 II-267, III-566
 Gómez-Villouta, Giglia I-1
 González-Abril, Luis II-470
 González-Castro, Víctor I-284
 Goodman, Nicholas I-590
 Graña, Manuel III-538
 Granmo, Ole-Christoffer III-199,
 III-209
 Griffin, J.D. III-528
 Grundy, John I-358
 Guedes, Frederico II-357
 Guerrero, José Luis III-498
 Guihaire, Valérie I-21
 Guilherme, Ivan Rizzo II-308
 Guo, Songshan I-31, III-179
 Gutierrez, P.A. III-1
 Guzmán-Martínez, R. I-284

 Häberle, Tilmann II-327
 Hakura, Jun III-419
 Hamiez, Jean-Philippe I-1
 Hao, Jin-Kao I-1, I-21
 Hayashi, Yuki III-239
 Hejazi, Hana I-388
 Hernández, Carmen III-538
 Hernández-Igüeño, Manuel I-489
 Hernández, Josefa Z. II-18
 Hernández-Orallo, José I-520
 Herrera, Francisco I-601, II-228
 Herrera-Viedma, Enrique III-429,
 III-450
 Hervás-Martínez, C. III-1, III-646
 Hinoshita, Wataru III-42
 Hirata, Norifumi II-525
 Hmer, Ali II-298
 Hofmann, Martin II-327
 Holubowicz, Witold I-671
 Hoonakker, Frank II-318
 Hou, Wen-Juan I-235
 Hsu, Jane Yung-Jen I-417

- Hsu, Kuo-Chung I-417
 Huang, Zhihu III-331
 Huete, Juan F. III-478
 Ibañez, Mario II-460
 Iburguren, A. III-508
 Ichise, Ryutarō II-545
 Imran, Hazra II-1
 Imura, Jun-Ichi I-62
 Ince, Gökhan I-62
 Ingaramo, Diego I-550
 Isern-Deyà, Andreu Pere I-185
 Ishibashi, Satoshi I-337
 Itoyama, Katsutoshi III-249
 Izuta, Guido I-499
 Jacquenet, François II-367
 Jaime-Castillo, Sergio III-291
 Jayaraman, Prem Prakash III-260
 Jerez, José Manuel I-317
 Jesús, María José del I-205
 Jiménez, Esther III-291
 Jo, Kang-Hyun III-379
 Julián, Vicente III-556
 Jung, Jason J. II-39
 Jurado-Lucena, Antonio I-427
 Jurio, Aranzazu III-341
 Kantardzic, Mehmed II-77
 Khodr, Hussein M. I-731
 Kim, Keehyung I-651
 Klein, Michel C.A. II-565
 Klippel, Alexander III-321
 Koh, Jia-Ling II-514
 Kojiri, Tomoko III-239
 Komatani, Kazunori I-102, II-585,
 III-249
 Kozielski, Stanisław III-616
 Kozik, Rafał I-671
 Krygowski, Filip II-173
 Kumova, Bora I. II-28
 Kuo, Jian-Long III-271
 Kurematsu, Masaki III-419
 Kybartas, Rimantas II-47
 Lachiche, Nicolas II-318
 Lama, Manuel I-175
 Lamirel, Jean-Charles III-139
 Larrañaga, Pedro I-531, III-149
 Lasso, M. III-576
 Lawrence, Elaine II-377, II-387, III-72
 Lawryńczuk, Maciej III-52
 Layeb, Abdesslem II-136
 Lázaro, Jesús III-281
 Ledezma, Agapito III-82
 Leguizamón, G. III-576
 Leng, Jinsong III-331
 Levin, Mark Sh. II-277
 Li, Dongguang III-331
 Lim, Andrew I-31, III-179
 Limeira, Giorgio O. II-357
 Lin, Frank Yeong-Sung III-189
 Lin, Gu-Yang I-417
 Li, Sheng-Yang I-235
 Liu, Shiang-Tai I-164
 Liu, Ziyang III-92, III-102
 Loia, Vincenzo III-460
 Lombardi, Leonardo O. I-306
 López-Molina, Carlos III-369
 Lowell, Daniel II-643
 Lozano-Tello, Adolfo I-661
 Luaces, Oscar II-337
 Lu, Ching-Hu I-417
 Lujak, Marin I-721
 Luna-Rodríguez, Juan-Jesús I-489
 Luong, Hoang N. II-126
 Luque Ruiz, Irene III-229
 Machado, K.S. III-119
 Madrid-Cuevas, F.J. III-350
 Maezawa, Akira III-249
 Mahanti, Prabhat K. II-484
 Makui, Ahmad I-254
 Małysiak-Mrozek, Bożena III-616
 Mandl, Monika I-621, I-631, I-641
 Markowska-Kaczmar, Urszula II-173
 Marques, Albino I-143
 Martín-Díaz, Ricardo I-489
 Martinelli, Francesco I-721
 Martínez-Álvarez, Francisco I-378
 Martínez, Ana M. I-570
 Martínez-Jiménez, Pilar I-489
 Martínez, Luis III-409
 Martínez Madrid, Natividad II-460
 Martínez-Marchena, Ildefonso III-606
 Martínez-Otzeta, J.M. III-508
 Martin, Florent I-468
 Martín, Jaime II-460
 Martín, José Luis III-281
 Martyna, Jerzy III-626
 Marusak, Piotr M. II-222
 Marwala, Tshilidzi III-62

- Matos, Pablo F. I-306
 Matsuyama, Kyoko II-585
 Matthews, Manton I-590
 Maudes, Jesús II-87, II-106
 Mazaira, L.M. III-159
 McKay, RI (Bob) I-651
 McKenzie, Amber I-590
 Medina-Carnicer, R. III-350
 Medina, Juan Miguel III-291
 Meger, Nicolas I-468
 Mehrotra, Kishan G. II-605
 Melo-Pinto, Pedro III-341
 Menéndez-Mora, Raúl Ernesto II-545
 Menezes, Hélio B. II-357
 Merchán-Pérez, Ángel III-112
 Meshoul, Souham II-136
 Mesiar, R. III-399
 Metta, Giorgio I-133
 Miró-Julιά, Margaret I-185
 Mizumoto, Takeshi I-102
 Mohammad, Yasser I-92
 Mohan, Chilukuri K. II-605
 Molina, José Manuel I-397, III-470,
 III-498
 Mollineda, Ramón A. I-541
 Montero, Miguel Á. I-611
 Montiel Sánchez, Ignacio I-427
 Mora, Juan III-596
 Morales-Bueno, Rafael I-560
 Morales, Juan III-112
 Mora-Lopez, Llanos III-596, III-606
 Moreira Bernardino, Anabela II-153,
 II-267
 Moreira Bernardino, Eugénia II-153,
 II-267
 Moreno-Muñoz, Antonio I-489
 Moro-Sancho, Q. Isaac II-96
 Motto Ros, Paolo III-11
 Mouhoub, Malek II-298
 Mourelle, Luiza de Macedo II-143,
 II-164
 Mrozek, Dariusz III-616
 Mucientes, Manuel I-175
 Muñoz, C. III-159
 Muñoz, Pablo I-72
 Muñoz-Salinas, R. III-350

 Nakadai, Kazuhiro I-51, I-62, I-102
 Nakamura, Masato II-535
 Natale, Lorenzo I-133

 Navarro, Karla Felix III-72
 Navarro, Martí III-556
 Nebot, Victoria II-504
 Nedjah, Nadia II-143, II-164
 Neves-Jr., Flávio III-546
 Nguyen, Hai T.T. II-126
 Nicolau, Guillermo III-21
 Nishida, Toyooki I-92, I-337
 Nkambou, Roger II-438
 Novoa, Clara II-643

 Ocon, Jorge I-154
 Ogata, Tetsuya I-102, II-585, III-42,
 III-249
 Okada, Shogo I-337
 Okuno, Hiroshi G. I-51, I-102, II-585,
 III-42, III-249, III-311
 Olivares, Joaquín III-301, III-389
 Olivas, Jose A. III-429
 Olivencia Polo, Fernando Agustín I-458
 Onaindia, Eva I-244
 Oommen, B. John III-209
 Orciuoli, Francesco III-460
 Ortega, Juan Antonio II-470
 Ortiz-Boyer, Domingo I-327
 Otsuka, Takuma I-102
 Ozono, Tadachika II-525, II-535, II-575,
 III-311

 Pagola, Miguel III-341, III-369
 Palacios, Rafael III-21
 Palomares, José M. III-301, III-389
 Palomar, Rafael III-389
 Pandolfi, D. III-576
 Pan, Youlian III-92, III-102
 Pardo, Carlos II-87, II-106
 Pardo, Thiago A.S. I-306
 Parras, Manuel I-205
 Pasero, Eros III-11
 Pasini, Francesco II-450
 Paternain, D. III-399
 Pathak, Shashank I-133
 Patricio, Miguel Á. I-397
 Pavón, Juan I-448
 Pedraza-Jimenez, Rafael III-488
 Pedraza, Juanita I-397
 Peña, José-María III-129
 Pérez-Godoy, María Dolores I-205
 Perez, Meir III-62
 Pérez, Pedro I-205
 Pérez-Vázquez, Ramiro III-478

- Petukhov, Maxim V. II-277
 Phan, Sieu III-92, III-102
 Piliougine, Michel III-596
 Plimmer, Beryl I-358
 Poirier, Pierre II-438
 Pomares, Héctor I-294, II-212, II-637
 Portilla-Figuera, J.A. III-1
 Potter, W.D. III-528
 Poyatos-Martínez, David I-427
 Prados, J.C. I-294
 Prieto, A. I-294
 Prieto, B. I-294
 Prieto, Óscar J. II-116
 Prodan, Ante I-113
 Provost, Michael I-123
 Puerta, José M. I-570, I-580
 Pulido, Belarmino II-116
 Pulina, Luca I-133
 Pum, Anton I-631
 Qin, Hu III-179
 Quevedo, José R. II-337
 Ramírez-Quintana, María José I-520
 Ramos, Carlos I-143, I-731
 Ramos-Muñoz, Iván II-96
 Rasconi, Riccardo I-154
 Raudys, Sarunas II-47
 Regazzoni, Carlo II-450
 Regueras, Luisa M. I-225
 Riquelme, José C. I-378
 Riva Sanseverino, Eleonora II-288
 Rivera, Antonio Jesús I-205
 R-Moreno, María D. I-72
 Rodellar, V. III-159
 Rodemann, Tobias I-62
 Rodrigues, Luiz Henrique A. II-337
 Rodríguez, Ángel III-112
 Rodríguez Cano, Julio C. III-478
 Rodríguez, Francisco I-264, I-294
 Rodríguez, José-Rodrigo III-112
 Rodríguez, Juan José II-87, II-106,
 II-116
 Rodríguez-Molins, Mario I-742
 Rodríguez, Sara I-407, III-556
 Rojas, Ignacio I-294, II-212, II-637
 Romero, Francisco P. III-429
 Rosso, Paolo I-550
 Rovira, Cristòfol III-488
 Rubin, David M. III-62
 Ruiz, D.D. III-119
 Ruiz-Morilla, Jose III-429
 Ruíz, Roberto I-611
 Ryu, Joung Woo II-77
 Sadi-Nezhad, Soheil I-254
 Sainz, Gregorio II-193
 Salcedo-Sanz, S. III-1
 Salehi-Abari, Amirali II-494
 Salido, Miguel Á. I-742, I-752, III-219
 Sánchez-Anguix, Víctor III-586
 Sanchez, David I-274
 Sánchez, José Salvador I-541
 Sánchez-Monedero, J. III-646
 Sánchez Montero, Ana María I-154
 Sanchez, Pedro J. III-606
 Sánchez-Pérez, Juan Manuel II-153,
 II-267, III-566
 Sanchis, A. III-1
 Sanchis, Araceli III-82
 Sanchiz, José Miguel I-368
 Santana, Roberto III-149
 Sanz-Bobi, Miguel A. III-21
 Saraiva do Nascimento Junior, Orlando
 II-308
 Sarro, Luis M. I-611
 Schippel, Stefan I-641
 Schlesinger, Federico I-701
 Schmid, Ute II-327
 Schneider, Thomas II-327
 Schubert, Monika I-621, I-631, I-641
 Schumann, Anika I-681
 Scott, Lesley E. III-62
 Sedano, Javier III-636
 Seepold, Ralf II-460
 Segovia-Vargas, M.J. III-1
 Serradilla, Francisco I-437
 Serrano-Cuerda, Juan I-348
 Serrano-Guerrero, Jesus III-429
 Shamsuddin, Siti Mariyam III-359
 Sharan, Aditi II-1
 Sharpanskykh, Alexei II-428
 Shearer, Heather III-92
 Shintani, Toramatsu II-525, II-535,
 II-575, III-311
 Shiramatsu, Shun II-525, II-535, II-575,
 III-311
 Siddiqui, Ghazanfar F. II-407
 Sidrach-de-Cardona, Mariano III-596,
 III-606
 Silva, António I-143

- Silva, Marcos Alexandre Rose I-215
 Simón-Hurtado, M. Aránzazu II-96
 Siqueira, Paulo Henrique II-247
 Soares, João P. I-731
 Soria-Morillo, L.M. II-470
 Soto, José M. III-301
 Soto, Ricardo I-82
 Stevens, Wendy III-62
 Stibor, Thomas I-509
 Strube de Lima, V.L. III-119
 Stütze, Thomas I-41
 Subirats, José Luis I-317
 Subramanian, Arun II-605
 Su, Kun Shian III-271
 Sun, Jigang II-615
 Swezey, Robin M.E. II-535
 Swirski, Konrad I-11

 Tacchella, Armando I-123, I-133
 Takahashi, Toru I-102, II-585, III-249
 Takasaki, Jun II-575, III-311
 Tamir, Dan E. II-643
 Tani, Jun III-42
 Tapia, Dante I. I-407
 Tchagang, Alain III-92
 Teixeira, Joaquim I-731
 Teppan, Erich I-641
 Tettamanzi, Andrea G.B. II-397, II-555
 Therón, Roberto II-12
 Torrecilla-Pinero, Fernando III-566
 Torrecilla-Pinero, Jesús A. III-566
 Treur, Jan II-407, II-428
 Triguero, Isaac I-601
 Trujillo, A.M. I-294
 Tsao, Jia-Hao I-235
 Tsujino, Hiroshi I-62
 Tubío, C. III-508

 Ul-Qayyum, Zia III-321
 Urda, Daniel I-317

 Valero, Soledad III-586
 Vale, Zita A. I-143, I-731
 Valiente-Rocha, Pablo A. I-661
 Vallez, Mari III-488
 Valls, Aida I-274
 Van den Poel, Dirk II-57
 Vanhoof, Koen I-691
 Van Tan, Vu I-478
 Vargas-Martínez, Adriana III-31
 Varnek, Alexandre II-318

 Varo-Martínez, Marta I-489
 Varshney, Pramod K. II-605
 Vasconcelos, Germano C. II-357
 Vavilin, Andrey III-379
 Vega-Rodríguez, Miguel Angel II-153,
 II-267, III-566
 Velasco, J.R. II-203
 Ventura, Sebastián III-439
 Verbiest, Nele III-450
 Verdú, Elena I-225
 Verdú, María Jesús I-225
 Victor, Patricia III-450
 Vidal, Juan Carlos I-175
 Vieira, Marina T.P. I-306
 Vieira, Petronio III-21
 Vieira, Rodrigo J.A. III-21
 Villagra, A. III-576
 Villar, José Ramón III-636

 Wagner, Alain II-318
 Walgampaya, Chamila II-77
 Wang, Xi II-627
 Wang, Yong I-358
 Wan, Wei II-418
 Warchol, Michal I-11
 Watanabe, Toyohide III-239
 Weber, Jörg I-711
 Wets, Geert I-691
 White, Tony II-494
 Wilson, Nic I-681
 Winck, A.T. III-119
 Wojdan, Konrad I-11
 Wotawa, Franz I-711
 Wu, Chien-Liang II-514
 Wu, Ying II-257

 Yamamoto, Lia III-546
 Yang, Jing-Hsiung III-271
 Yazidi, Anis III-209
 Yilmaz, Nihat II-47
 Yi, Myeong-Jae I-478
 Yoshida, Takami I-51
 Yuan, Zhi I-41

 Zafra, Amelia III-439
 Zaslavsky, Arkady III-260
 Zhang, Huidong I-31
 Zhang, Zizhen III-179
 Zhu, Wenbin I-31
 Zidrasco, Tatiana II-575, III-311
 Zuloaga, Aitzol III-281